**STMICROELECTRONICS**

# COMMON CRITERIA FOR IT SECURITY EVALUATION

# ST33TPM12LPC

## SECURITY TARGET – PUBLIC VERSION

OCTOBER 2012

### Document Identification

| Division | Department | Author | Confidentiality level |
|---|---|---|---|
| SMD | Secure Software Solutions | Olivier Collart | Public |

### Document Revision

| Version | Date | Author | Modifications |
|---|---|---|---|
| 02-01 | 01/10/2012 | Olivier Collart | Security Target for evaluation |
| 02-01p | 01/10/2012 | Olivier Collart | Security Target Lite for public release |

# Table of Contents

# List of Tables

## List of Figures

# 1 INTRODUCTION

## 1.1 ST Reference

This security target is referenced with the following information

- Filename: ST33TPM12LPC_ST (02.01p)
- Revision: 02.01p, issued October 2012
- Registration: registered at STMicroelectronics under reference ST33TPM12LPC_ST_10_001_V02.01p

This security is strictly conformant to the TPM Protection Profile [TPM1.2 PP rev116].

## 1.2 Purpose

This document presents the Security Target (ST) of the ST33TPM1.2 product.

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

A list of abbreviation definition is provided in Appendix A

## 1.3 Context

The original text from the protection profile [TPM1.2 PP rev116] is indicated with a light grey font.

# 2 TOE DESCRIPTION

## 2.1 TOE reference

This security target covers three targets of evaluation:

**Table 1: Target of evaluations references**

| TOE | Firmware revision | Product ordering information | Field Upgradable firmware binary |
|---|---|---|---|
| TOE 1 | 0x01 0x02 0x0D 0x00 | ST33ZP24xxxxPVSC | |
| TOE 2 | 0x01 0x02 0x0D 0x08 | ST33ZP24xxxxPVSH | FU_12D8.fu |
| TOE 3 | 0x01 0x02 0x0D 0x0C | ST33ZP24xxxxPVSP | FU_12DC.fu |

The package information is defined by the “xxxx” field in the product ordering information.

The package is not part of the TOE.

The TOE 1 can be upgraded to TOE 2 or 3 by loading the firmware binary 1.2.D.8 or 1.2.D.C respectively.

The TOE2 can be upgraded to TOE 3 by loading the firmware 1.2.D.C.

Once a firmware is loaded it is not possible to load a firmware with a lower revision.

## 2.2 TOE Overview

### 2.2.1 TOE Definition

The TOE is the TCG PC Client Specific Trusted Platform Module (PCCS TPM). This TPM is hardware, firmware and/or software that implements the functions defined in the TCG Trusted Platform Module Main Specification, version 1.2, [5] [6] [7] and the PC client specific interface specification [8]. The TCG Trusted Platform Module Specification describes the design principles [5], the TPM structures [6] and the TPM commands [7]. The PC Client Interface Specification [8] describes the platform-specific set of requirements of the TPM for the PC Client, the details of what interfaces and protocols are used to communicate with the TPM and specific items like the minimum number of PCRs required and NV Storage available.

The primitives provided by the TOE include cryptographic algorithms for key generation, digital signatures, random number generation, sealing data to system state, protected storage, binding information to the TPM, support of direct anonymous attestation and physical protection. Attestation by the TOE is an operation that provides proof of data known to the TPM. This is done by digitally signing specific internal TPM data using an Attestation Identity Key (AIK). The acceptance and validity of both the integrity measurements and the AIK itself are determined by the Verifier. The AIK is obtained using either the Privacy Certification Authority or the Direct Anonymous Attestation (DAA) protocol. The DAA is a protocol for vouching for an AIK using zero-knowledge-proof technology.

### 2.2.2 TOE Major Security features

The PCCS TPM provides all services required for a TPM in the TCG Trusted Platform Module Main Specification, version 1.2, [5] [6] [7] and additional services that are optional in the main TPM specification but mandatory in the PC client specific interface specification [8]. The PCCS TPM provides physical protection for internal user data and TSF data.

In TCG systems roots of trust are components that must be trusted because misbehavior might not be detected. There are commonly three Roots of Trust in a trusted platform; a root of trust for measurement (RTM), root of trust for storage (RTS) and root of trust for reporting (RTR). The RTM is a computing engine capable of making inherently reliable integrity measurements. Typically the normal platform computing engine is controlled by the core root of trust for measurement (CRTM). The CRTM is the instructions executed by the platform when it acts as the RTM. The RTM is also the root of the chain of transitive trust. The RTS is a computing engine capable of maintaining an accurate summary of values of integrity digests and the sequence of digests. The RTR is a computing engine capable of reliably reporting information held by the RTS. The TCG Specification Architecture Overview [11] provides a more detailed description.

### Support for the Root of Trust for Measurement

The TPM supports the integrity measurement of the trusted platform by calculation and reporting of measurement digests of measured values. The measurement values are representations of embedded data or program code scanned and provided to the TPM by the measurement agent, such as the Root-of-Trust-for-Measurement. The TPM supports cryptographic hashing of measured values and calculates the measurement digest by extending the value of a Platform Configuration Register (PCR) with a calculated or provided hash value by means of the SHA-1. The PCRs are shielded locations of the TPM which can be reset by TPM reset or a trusted process, written only through measurement digest extensions and read.

### Root of Trust for Reporting

The root of trust for reporting (RTR) exposes the measurement digests stored in the PCRs and attests to the authenticity of these measurement digests based on trusted platform identities or the Direct Anonymous Attestation Protocol. The trusted platform identities for RTR are defined by Attestation Identity Credentials for Attestation Identity Keys (AIK) generated by the TPM. The TPM creates digital signatures over the PCR values using an Attestation Identity Key.

Each TPM is identified and validated using its Endorsement Key. A TPM has only one endorsement key pair. The Endorsement Key is transitively bound 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.

The Endorsement Key is used in the process of issuance the Attestation Identity Credentials and to establish a platform owner.

### Root of Trust for Storage

The TPM may be used to provide secure storage for an unlimited number of private keys or other data by means of encryption. The resulting encrypted file, which contains header information in addition to the data or key, is called a BLOB (Binary Large Object) and is output by the TPM and can be loaded in the TPM when needed. The functionality of the TPM can also be used so that private keys generated on the TPM can be stored outside the TPM (encrypted) in a way that allows the TPM to use them later without ever exposing such keys in the clear outside the TPM. The TPM uses RSA key technology to encrypt data and keys and may implement cryptographic algorithms of equivalent strength.

The functionality used to provide secure storage is:

- TPM_Seal and TPM_Unseal, which perform RSA encrypt and decrypt, respectively, on data that is externally generated. The sealing operation encrypts not only the data, but also the values of the selected PCRs and the locality that must exist during for unseal and tpmProof, which is a unique secret identifier for the TPM sealing the data. To unseal the data, three conditions must exist:

(i) the appropriate key must be available for unseal,

(ii) the TPM PCRs must contain the values defined at the time of the seal operation, and

(iii) 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 explicitly state the future "trusted" configuration that the platform must be in for the decrypted key to be used and for decryption to only occur on the specified TPM.

- TPM_Unbind, which RSA decrypts a blob created outside the TPM that has been encrypted using a public key where the associated private key is stored in the TPM.

The key types used for the Root for Trust of Storage include:

- The Storage Root Key (SRK), which is 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 owned TPM contains a SRK, generated by the TPM at the request of the Owner. Under that SRK may be organized different trees dealing with migratable data or non-migratable data.
- Storage keys, which are used to RSA encrypt and RSA decrypt other keys and sealed data with their security attributes in the Protected Storage hierarchy, only.
- Binding Keys, which are used for TPM_Unbind operations only. A binding 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.

### Other security services and features

The TPM provides cryptographic services hashing of arbitrary data by means of the hash function SHA-1 and creation of digital signatures with signing keys which must be a leaf of the Storage Root Key hierarchy. The private key of a singing key pair is used for signing operations only.

The TPM provides non-volatile storage as a shielded location for data of external entities.

The TPM owner controls access to the non-volatile storage. The access control may include the need for authentication of the user, delegations, PCR values and other controls. Keys managed by the TPM may be non-migratable, migratable or certifiable migratable. A non-migratable key is a key that cannot be transported outside beyond a specific TPM. A migratable key is a key that may be transported outside the specific TPM. In addition some keys must be bound to a specific TPM but should be able to be backed-up or migrated under certain circumstances. The certified migration allows a Migration Selection Authority therefore to control a migration process without handling the migrated key itself or respectively uses a Migration Authority to control the migration process without the knowledge of the data or the migrated key. Those keys which are intended for certified migration are called certifiable migratable keys

The TPM provides a "tick counter" as a count of the number of ticks that have occurred since the start of a timing session. The time between the ticks is identified via a "tick rate" but it is the responsibility of the caller to associate the ticks to an actual UTC time.

The TPM provides also a monotonic counter as an ever-increasing incremental value for external use.

Generation and import of the Endorsement key pair and certificate

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

The Endorsement key pair is generated by a HSM (Hardware Security Module) and then stored encrypted with a 2-DES transport key on a key server.

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

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

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

The STMicroelectronics intermediate certificates and the Certificate Practice Statement are available from the STMicroelectronics website: http://www.st.com/TPM/repository/

### 2.2.3 Supporting software non included in the TOE

The TPM is usually provided to platform integrators with

- Drivers for BIOS
- Drivers for different platforms OS (depending on OS)
  - In kernel mode and
  - in user mode

The TPM is usually provided to end users with

- Adapted drivers to platform configuration
- A “TPM Software Stack” (TSS) whose interfaces are also standardized by the TCG. The TSS provides services to ease the usage of the TPM by different application developers. It is among other things responsible for resource management.
- Various applications providing end-users services

**Note:** *Some OS include native drivers and part of the TSS services into their standard release.*

None of these software components are in the scope of the evaluation.

## 2.3 TOE Description

### 2.3.1 *TOE hardware description*

The TOE is based on the ST33ZP24 hardware platform based on the ST33 product family.

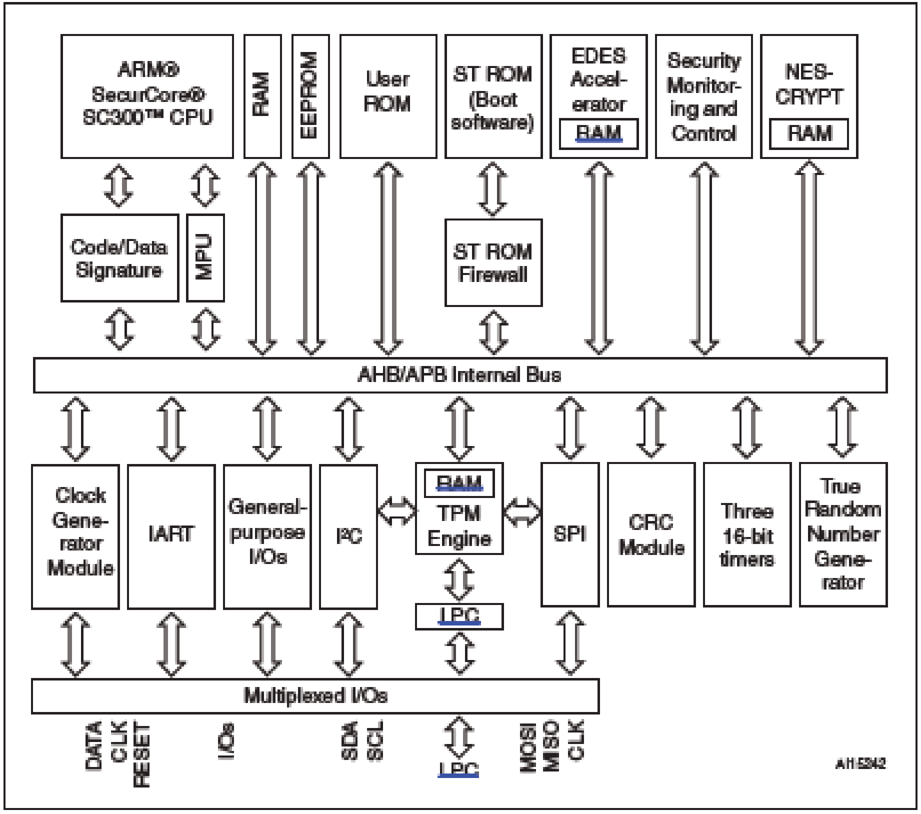
The ST33ZP24 is a serial access microcontroller designed for Trusted Platform Module applications that incorporates the most recent generation of ARM processors for embedded secure systems. Its SecurCore® SC300™ 32-bit RISC core is built on the Cortex™ M3 core with additional security features to help to protect against advanced forms of attacks.

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

The ST33ZP24 features 2 hardware accelerators for advanced cryptographic functions

- The EDES peripheral provides a secure DES (Data Encryption Standard) algorithm implementation,
- the NESCRYPT crypto-processor efficiently supports the public key algorithm.

**Figure 1: ST33ZP24 block diagram**



More precisely the TOE hardware includes the following features:

- A CPU ARM® SecurCore® SC300™ 32-bit RISC core is built on the Cortex™ M3 core
- CPU clock frequency up to 35MHz
- 140 Kbytes of User ROM

- 5 Kbytes of User RAM
- 3 Kbytes of transfer TPM RAM, 2 Kbytes usable for FIFO
- 24 Kbytes of EEPROM with 23 Kbytes for user and including a
  - 128 bytes User OTP area
  - 512 bytes for Monotonic counters
- Power saving standby state
- three 16-bit timers
- one 24-bit timer inside CPU

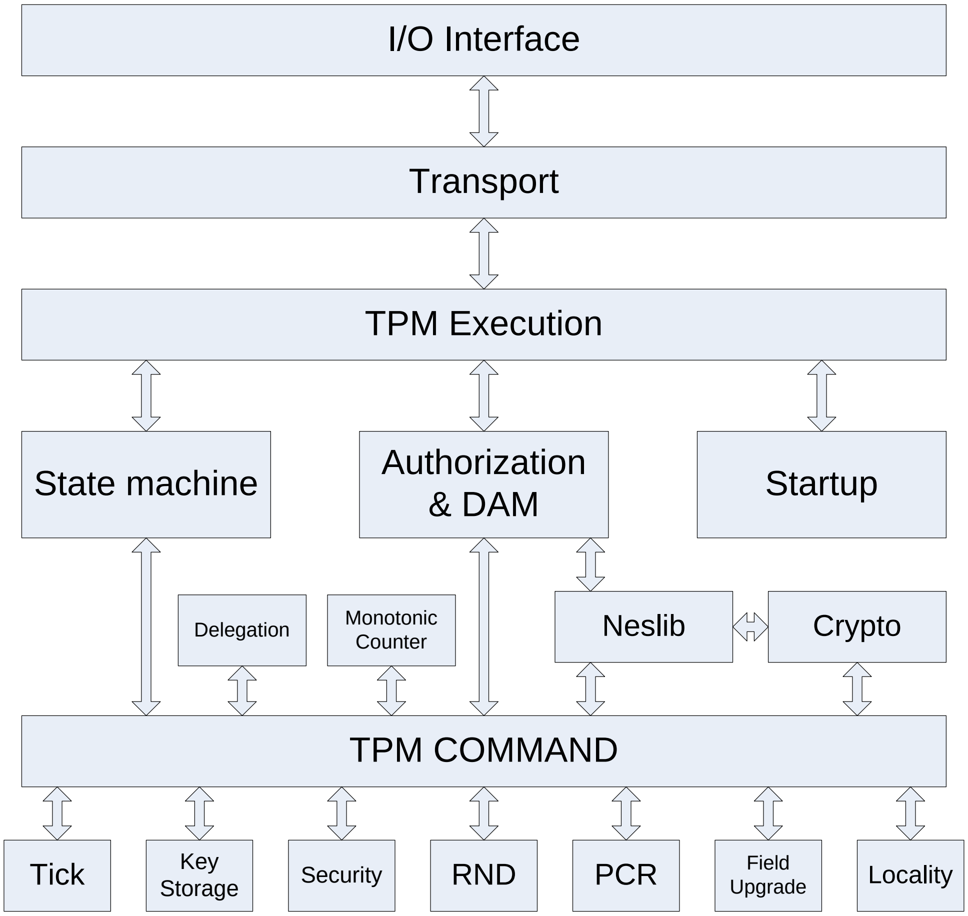
Moreover the ST33ZP24 hardware includes also the following security features:

- Active shield
- Memory protection unit (MPU)
- Monitoring of environmental parameters
- Code/Data Signature for Protection against fault attacks
- ISO 3309 CRC calculation block
- AIS-31 Class P2 compliant true random generator (TRNG)
- Hardware security-enhanced DES accelerator
  - Including 256 bytes of RAM
- NESCRYPT coprocessor for public key cryptography
  - Including 2 Kbytes of RAM
- The ST ROM is located in a zone of 20Kbytes ROM protected by a firewall. This ST ROM includes:
  - Test program used to validate the TOE production
  - Boot software: running at TOE start-up and controlling the correct behaviour of the TOE

### 2.3.2 *TOE Firmware description*

The ST33TPM1.2 firmware is an implementation compliant with the TPM specifications [5] [6] [7] and PCClient platform specifications [8].

**Figure 2: ST33TPM1.2 Firmware block diagram**



The firmware of the TOE includes the following modules

- I/O Interface: configured to support exclusively LPC protocol
- Transport: wrapper-unwrapper for transport sessions
- TPM Execution: dispatcher for TPM processing
- State Machine: TPM state machine
- Authorization & DAM: module managing the authorization and dictionary attack mitigation countermeasure
- Startup: module responsible for TOE test at startup and TOE self test
- TPM Command: processing of TPM commands
- Delegation: module responsible for delegation
- Monotonic counter: module responsible for management of monotonic counter logic
- Neslib: cryptographic library providing basic crypto services protected against fault attacks: RSA, SHA1, AES

- Crypto: modules providing crypto services: AES in CTR mode, HMAC; PKCS signature and encryption, MGF, key derivation function
- Tick: module providing services for Tick Counters management
- Key storage: module managing key repository
- PCR: module managing Platform Configuration Registers
- RND: Random Number generation services
- Security: security configuration
- Locality: locality management as defined in [8]
- Field Upgrade: module managing the secure download of additional code in the field. The code must be encrypted and signed with a RSA 2048-bit authentication key to modify the protected capabilities of the TOE.

## 2.4 TOE lifecycle

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

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

as defined in the PP [10] section 1.3.3.

The phase 1 that includes TPM development involves the sites of

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

for the hardware development activities and

- ST ROUSSET (FRANCE)
- ST ZAVENTEM (BELGIUM)
- ST PRAHA (CHECK REPUBLIC)

for the embedded software development activities.

The phase 2 that includes the TPM manufacturing, the TPM conformance testing, the TPM-Mfg EK key pair download and the TPM-Mfg EK credential issuance involves the sites of

- ST ROUSSET (FRANCE)
- ST TOA PAOH.( SINGAPORE)

# 3 CONFORMANCE CLAIM

## 3.1 CC Conformance Claim

This security target is **conformant** to the Common Criteria version 3.1 R3.

This security target claims to be Common Criteria version 3.1 R3

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

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

## 3.2 PP Claim

This security target is in **strict conformance** to the Protection Profile PC Client Specific TPM, Family 1.2 Level 2 Revision 116 (Version 1.2) released by the Trusted Computing Group dated 18 May 2011.

The protection profile is registered and **certified through a assurance continuity maintenance report** by the Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0030-2008-MA-01, , dated 6 October 2011.

## 3.3 Package claim

This ST is conforming to assurance package EAL4 augmented with

- ALC_FLR.1
- AVA_VAN.4

defined in CC Part 3 [CCMB-2009-07-004].

## 3.4 Conformance Rationale

This security target claims **strict conformance** to only one PP [TPM1.2 PP rev116].

The Target of Evaluation (TOE) is a complete solution implementing the TCG Trusted Platform Module main specifications Version 1.2 level 2 revision 116 ([TPM Part1 116], [TPM Part2 116] and [TPM Part3 116]) and the TCG PC Client Specific TPM Interface Specification, Version 1.21 Final, Revision 1.00 ([PC Client TIS 1.21]) as defined in the PP section 1.3.1. So the TOE is **consistent** with the **TOE type** in the PP.

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

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

The **security requirements** of this security target are **consistent** with the statement of the security requirements in the PP as the security target claims strict conformance to the PP. One security functional requirement is added in this security target to cover the authenticity verification of the field upgrade binary file. All assignments and selections of the security functional requirements are done in the PP section 6.1 and in this security target section 7.1.

# 4 Security Problem Definition

The content of the PP [TPM1.2 PP rev116] applies to this chapter completely.

## 4.1 Threats

The threats are defined in the PP section 4.1, no other threats are added.

The primary assets that have to be protected by the TOE against the listed threats are:

- User data
- TSF Data.

In order to protect the access to the primary assets, the following components are also considered are assets:

- Hardware of TOE
- Embedded firmware.

## 4.2 Organizational Security Policies

The organizational security policies are defined in the PP section 4.2, no other organizational security policies are added.

## 4.3 TOE Operational environment assumptions

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

The TOE assumptions to the IT environment are defined in the PP section 4.3, no other assumptions are added.

# 5 SECURITY OBJECTIVES

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

## 5.1 Security Objectives for the TOE

The security objectives of the TOE are defined and described in the PP, section 5.1, no other security objectives are added.

## 5.2 Security Objectives for the operational environment

The security objectives for the operational environment are described in the PP, section 5.2, no other security objectives are added.

## 5.3 Security Objectives rationale

The security objectives rationale is described in the PP, section 5.3., no other security objectives are added.

# 6 EXTENDED COMPONENT DEFINITION

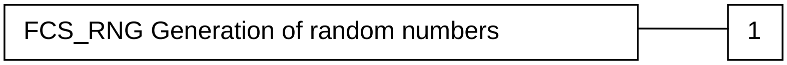
The extended component “FCS_RNG Generation of random numbers” (FCS_RNG.1) is already described in the PP. No other extended components are added.

### FCS_RNG Generation of random numbers

**Family behaviour**

This family defines quality requirements for the generation of random numbers which are intended to be use for cryptographic purposes.

**Component levelling**:



FCS_RNG.1 Generation of random numbers requires that random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: ZCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid*] random number generator that implements: [assignment: *list of security capabilities*].

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

## 7 SECURITY REQUIREMENTS

For this section the PP section 6 can be applied with the following exceptions.

### 7.1 Security Functional Requirements for the TOE

The security functional requirements (SFRs) for the TOE are defined in the PP section 6.1. All assignments and selections of the Security Functional Requirements are done in the PP with the exception of the following SFRs that required to be completed in the security target.

The operations completed in this security target are marked in *italic* font.

**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.*

**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 *authentication data of the user using OperatorAuth, TPM Owner, delegation entities, owner of entities, user of entities* when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use roles defined in [TPM Part2 116] and [TPM Part3 116] when interpreting the TSF data from another trusted IT product.

**FCS_RNG.1** **Random number generation**

Hierarchical to: No other components

Dependencies: No dependencies

FCS_RNG.1.1 The TSF shall provide a *physical* random number generator that implements: *total failure test of the random source*.

FCS_RNG.1.2 The TSF shall provide random numbers that meet *AIS31 Class P2*.

**FCS_CKM.1 /AES** **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 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*

**FCS_CKM.4** **Cryptographic key destruction**

Hierarchical to: No other components

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

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method: *key overwriting with "0"* or *generation of new key encryption key* that meets the following: *none*

**FCS_COP.1 /RSA_Sig** **Cryptographic operation**

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 /RSA_Sig The TSF shall perform signature generation and signature verification in accordance with a specified cryptographic algorithm *RSA signature scheme [5] section 31.2.1, 31.2.2, 31.2.3* and cryptographic key sizes *RSA 512, 1024, 2048* that meet the following: *PKCS#1 V2.0* ([PKCS#1])

**FCS_COP.1 /VAF** **Cryptographic operation**

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 /VAF The TSF shall perform verification  of the signature attached to the Field Upgrade firmware in accordance with a specified cryptographic algorithm SHA1 and PKCS#1 *RSA signature* and cryptographic key size of *RSA 2048 bits* that meet the following: RSASSA-PKCS1-v1.5 ([PKCS#1])

**FCS_COP.1 /RSA_Enc** **Cryptographic operation**

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 /RSA_Enc The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm *RSA encryption scheme [5] sections 31.1.1 and 31.1.2* and cryptographic key sizes *RSA 512, 1024, 2048* that meet the following: *PKCS#1 V2.0* [PKCS#1].

**FCS_COP.1 /SymEnc2** **Cryptographic operation**

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 /SymEnc2 *The TSF shall perform symmetric encryption and decryption in accordance with a specified cryptographic algorithm AES mode CTR as described in [5] section 31.1.3 and cryptographic key size of 128 bits that meet the following: FIPS PUB 197, 2001 November 26*

**FIA_UID.1** **Timing of identification**

Hierarchical to: No other components

Dependencies: No dependencies

FIA_UID.1.1 The TSF shall allow:

1. to execute commands indicated in PP [TPM1.2 PP rev116] table 12 column RQU 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. to execute the commands TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END

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

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

**FIA_UAU.1** **Timing of authentication**

Hierarchical to: No other components

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1 The TSF shall allow

1. to execute commands indicated in table 12 column RQU 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. to execute the commands TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END

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

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

**FIA_UAU.5** **Multiple authentication mechanisms**

Hierarchical to: No other components

Dependencies: No dependencies

FIA_UAU.5.1 The TSF shall provide

1. OIAP authorization session,
2. OSAP authorization session,
3. DSAP authorization session,
4. Transport session,
5. Commands which require authorization and are executed outside a authorization session

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the following rule: *the TOE maintains a counter of unsuccessful authentication. This counter is incremented each time a wrong authentication value is provided for OIAP, OSAP, DSAP authorization sessions. When the counters reaches a specific value (40dec), the activation Flag is set to True which makes that the response will be delayed. For each new unsuccessful authentication the response time is doubled until a maximum value. When this maximum value is reached the response time is always the maximum value*

**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 *40* unsuccessful authentication attempts occur 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. *Add a delay in command response time for authorized commands,*

*non authorized commands time is kept unchanged. This forbids another authorized command to be processed. Each new failed authentication doubles the response time up to a maximum value.*

**FDP_ACF.1/ Deleg** **Security attribute based access control**

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

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: if the TPM command is listed in the table at [6], section 20.2.1 “Owner Permission Settings” including TPM_KeyControlOwner[1] at bit 31 or the key is listed in the table at [6] section 20.2.3 “Key Permission Settings”, then the TPM owner or the key user can delegate that capability to a trusted process.

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

1. if the TPM command is listed in the table at [6], section 20.2.2 “Owner commands not delegated”
2. if the key is listed in the table at [6], section 20.2.4 “Key commands not delegated”, then the command can not be delegated**.**
3. The delegation is denied if family linked to delegation row, delegation owner blob or delegation key blob flag TPM_FAMFLAG_ENABLED is set to false
4. The delegation family configuration is no more editable when TPM is unowned if family flag TPM_DELEGATE_ADMIN_LOCK is set to TRUE

**FDP_ACF.1/ KeyMan** **Security attribute based access control**

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

[1] TPM_KeyControlOwner at bit 31 in section 20.2.1 “Owner Permission Settings” was added in revision 116.

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 SRF related security attributes disableOwnerClear anddisableForceClear of the TOE,
    - (c) User keys with the security attributesauth<DataUsage, keyUsage, keyFlags, and ownerEvict,
    - (d) Wrapped Key Blob with the 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 user “World” is allowed to create an EK if the EK does not exist already.
2. The user “World” is allowed to read the public part of an EK if the TOE is unowned.
3. The TPM owner is allowed to read the public part of an EK.
4. The user “World” 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 user “World” 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 an User Key and export this User key wrapped with the key the owns key except this entity owner is not the TPM owner and the key to 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.
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 to use the command to be executed,
    - (d) the security attribute PCRInfo of the User Key object does not allow to use the object in the actual 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 authorise access of subjects to objects based on the following additional rules:

1. *The execution of the commands TPM_CreateWrapKey, TPM_LoadKey, TPM_LoadKey2, TPM_Unseal, TPM_GetPubKey, TPM_CertifyKey, TPM_CertifyKey2, TPM_MakeIdentity, TPM_DSAP, TPM_ChangeAuthAsymStart, TPM_CMK_CreateKey, TPM_CMK_SetRestrictions, TPM_CMK_CreateBlob and TPM_CMK_ConvertMigration depends on the values of the security attribute TPM_KEY_FLAG (keyFlags).*
2. *The execution of the commands TPM_CMK_SetRestrictions, TPM_ChangeAuthAsymStart, TPM_Take_Ownership, TPM_Seal, TPM_Unseal,, 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_Tick StampBlob and TMK_EstablishTransport depends on the values of the security attribute TPM_KEY_USAGE (KeyUsage).*
3. *The execution of the commands TPM_Seal, TPM_Unseal, TPM_LoadKey, TPM_LoadKey2, TPM_MakeIdentity, TPM_GetPubKey, TPM_CertifyKey, TPM_CertifyKey2, TPM_CMK_CreateKey, TPM_NV_WriteValue, TPM_NV_WriteValueAuth, and TPM_NV_ReadValueAuth depends on the values of the security attribute pcrInfo .*
4. *The execution of the commands TPM_TakeOwnership, TPM_AuthorizeMigrationKey, TPM_CMK_CreateTicket and TPM_CMK_CreateBlob depends on the values of the security attribute algorithmParams.*
5. *The execution of the commands TPM_Startup, TPM_KeyControlOwner, TPM_FlushSpecific and TPM_EvictKey depends on the values of the security attribute OwnerEvict.*

FDP_ACF.1.4/ KeyMan

The TSF shall explicitly deny access of subjects to objects based on the rule: *none*

**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 *no role* to specify alternative initial values to override the default values when an object or information is created.

**FDP_ACF.1 /MigK**

**Security attribute based access control**

Hierarchical to: No other components

Dependencies: FDP_ACC.1 Subset of access control

FMT_MSA.3 Static attribute initialisation

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 migrationSheme,
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 migrationKeyAuth.

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 migratable User key is allowed to create a Wrapped Key Blob for this migratable key by means of the command TPM_CMK_CreateKey, if it is authorized for 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 command TPM_CreateMigrationBlob.
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 command TPM_CMK_CreateBlob.
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 command TPM_MigrateKey,
5. The Entity owner of the private part of migration User key is allowed to convert a Migration Key Blob by means of the command TPM_ConvertMigrationBlob and a Certified Migration Key Blob by means of the command TPM_CMK_ConvertMigration 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 execution of the commands TPM_CreateMigrationBlob, TPM_ConvertMigrationBlob, TPM_CMK_CreateKey, TPM_CMK_CreateBlob and TPM_CMK_ConvertMigration depends on the value of the security attribute payload type.*

*2. The execution of the commands TPM_CreateMigrationBlob, and TPM_CMK_CreateBlob depends on the value of the security attribute migrationKeyAuth.*

3. *The execution of the command TPM_CMK_CreateKey depends on the value of the security attribute migrationAuthorityApproval.*

FDP_ACF.1.4 /MigK

The TSF shall explicitly deny access of subjects to objects based on the: *none.*

**FDP_ACF.1 /M&R**

**Security attribute based access control**

Hierarchical to: No other components

Dependencies: FDP_ACC.1 Subset of access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 /M&R

The TSF shell enforce the Measurement and Reporting SFP to objects based on the following:

1. Subjects:
    a. SHA-1 session,
    b. user with the security attribute locality,
    c. entity owner of the signature key with the security attribute usageAuth,
2. Objects:
    a. PCR with the security attribute pcrReset, pcrResetLocal, pcrExtend-Local
    b. Signature key with the 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 command TPM_SHA1Start.
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 command TPM_SHA1Update.
3. 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 output the hash value by command TPM_SHA1Complete.
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 command TPM_SHA1CompleteExtend.
5. If the pcrReset is TRUE the command TPM_Startup is allowed to set a PCR to 0xFF…FF.
6. If the pcrReset is FALSE the command TPM_Startup 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, than the command TPM_PCR_Reset 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 pcrExtend-Local of the selected PCR the command TPM_SHA1CompleteExtend 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 selected PCR with the final digest of the SHA-1 session.
9. If the user presents the locality matching the security attribute pcrExtend-Local of the selected PCR the command TPM_Extend 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 command TPM_PCRRead.

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 command TPM_Quote or TPM_Quote2.
12. The user "World" under locality 4 is allowed to execute the LPC commands TPM_HASH_START, TPM_HASH_DATA ans TPM_HASH_END.
13. Additional rules for operations, based on security attributes of the subjects and objects: *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 execution of the command TPM_PCR_Reset depends on the values of the security attributes pcrReset and pcrResetLocal.*
2. *The execution of the commands TPM_SHA1CompleteExtend and TPMExtend depends on the value of the security attribute pcrExtendLocal.*
3. *The execution of the commands TPM_Quote and TPM_Quote2 depends on the value of the security attribute KeyUsage.*

FDP_ACF.1.4 /M&R

The TSF shall explicitly deny access of subjects to objects based on the rules*: none.*

**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 *no role* to specify alternative initial values to override the default values when an object or information is created.

**FDP_ACF.1 /NVS**

**Security attribute based access control**

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

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 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 command TPM_NV_DefineSpace 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 command TPM_NV_DefineSpace.
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 command TPM_NV_WriteValue if
    a. TPM_NV_PER_OWNERWRITE is TRUE,
    b. the user match the requirement for physical presence defined in TPM_NV_PER_PPWRITE,
    c. the locality of the user mach the localityAtRelease defined for the TPM_NV_DATA_AREA and
    d. if pcrInfWrite defines a PCR selection the actual values of the selected
    e. PCR shall match the digestAtRelease in pcrInfoWrite.
5. The entity owner is allowed to write an NV storage area by means of the command TPM_NV_WriteValueAuth 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. if 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 command TPM_NV_ReadValue if
    a. TPM_NV_PER_OWNERREAD is TRUE,
    b. the user match 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. if 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 command TPM_NV_ReadValueAuth 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. if 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. *The value of security attribute* ***nvLocked*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_WriteValue and TPM_NV_ReadValue.*
2. *The value of security attribute* ***noOwnerNVWrite*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_WriteValue.*
3. *The value of security attribute* ***pcrInfoRead*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_ReadValue and TPM_NV_ReadValueAuth.*
4. *The value of security attribute* ***pcrInfoWrite*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_WriteValue and TPM_NV_WriteValueAuth.*
5. *The value of security attribute* ***localityAtRelease*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_ReadValue, TPM_NV_ReadValueAuth, TPM_NV_WriteValue and TPM_NV_WriteValueAuth.*
6. *The value of security attribute* ***TPM_NV_PER_AUTHREAD*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_ReadValue and TPM_NV_ReadValueAuth.*
7. *The value of security attribute* ***TPM_NV_PER_AUTHWRITE*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_WriteValue and TPM_NV_WriteValueAuth.*
8. *The value of security attribute* ***TPM_NV_PER_OWNERREAD*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_ReadValue and TPM_NV_ReadValueAuth.*
9. *The value of security attribute* ***TPM_NV_PER_OWNERWRITE*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_WriteValue and TPM_NV_WriteValueAuth.*
10. *The value of security attribute* ***TPM_NV_PER_PPREAD*** *impacts the processing of the following commands: TPM_NV_ReadValue and TPM_NV_ReadValueAuth*
11. *The value of security attribute* ***TPM_NV_PER_PPWRITE*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_WriteValue and TPM_NV_WriteValueAuth.*
12. *The value of security attribute* ***TPM_NV_PER_READ_STCLEAR & bReadSTClear*** *impacts the processing of the following commands: TPM_NV_ReadValue and TPM_NV_ReadValueAuth.*
13. *The value of security attribute* ***TPM_NV_PER_WRITE_STCLEAR & bWriteSTClear*** *impacts the processing of the following commands: TPM_NV_WriteValue and TPM_NV_WriteValueAuth.*
14. *The value of security attribute* ***TPM_NV_PER_WRITEDEFINE & bWriteDefine*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_WriteValue and TPM_NV_WriteValueAuth.*
15. *The value of security attribute* ***TPM_NV_PER_GLOBALLOCK*** *impacts the processing of the following commands: TPM_NV_DefineSpace, TPM_NV_WriteValue and TPM_NV_WriteValueAuth.*

FDP_ACF.1.4 /NVS

The TSF shall explicitly deny access of subjects to objects based on the rules:

1. If TPM_NV_PER_READ_STCLEAR is TRUE the NV storage area can not 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 can not 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 can not be written after performing the TPM_NV_DefineSpace command and one successful write.
4. If TPM_NV_PER_GLOBALLOCK is TRUE the NV storage area can not be written after successful write to index 0 until TPM_Startup(ST_Clear).
5. *the access to command TPM_NV_DefineSpace is denied if*
    - *a. TPM_NV_PER_OWNERWRITE and TPM_NV_PER_AUTHWRITE are both set to TRUE or*
    - b. *TPM_NV_PER_OWNERREAD and TPM_NV_PER_AUTHREAD are both set to TRUE*

**FDP_ACF.1 /MC** **Security attribute based access control**

Hierarchical to: No other components

Dependencies: FDP_ACC.1 Subset of access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 /MC

The TSF shall enforce the Monotonic Counter SFP to objects based on the following:

1. Subjects: TPM owner, Entity owner of the monotonic counter object, OSAP session, DSAP session,
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 authorise access of subjects to objects based on the following additional rules:

1. the execution of the commands TPM_IncrementCounter, TPM_ReadCounter, TPM_ReleaseCounter and TPM_ReleaseCounterowner depends on the value of the security

attribute countID

FDP_ACF.1.4 /MC

The TSF shall explicitly deny access of subjects to objects based on the rule:

1. The TSF shall disallow the operation read or increment the monotonic counter if the countID is invalid.

**FDP_ACF.1 /EID**

**Security attribute based access control**

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

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 have the security attribute TPM_KEY_STORAGE and is not migratable,
    b. the security attributes pcrInfo of sealed data blob shall match to the values in the PCR indicated by pcrInfo,
    c. the security attributes tmpProof of sealed data blob shall match to 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 match 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:

1. *The execution of the command TPM_Unseal depends on the value of the 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: *none*

**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 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 /M&R The TSF shall allow *no role* to specify alternative initial values to override the default values when an object or information is created.

**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: *no role* to specify alternative initial values to override the default values when an object or information is created.

**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, ,*authorization (dictionary attack) and failure of operating conditions (power supply)* .

**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.

## 7.2 Security Assurance Requirements

The security assurance requirements (SAR) of the TOE are the assurance components of the Evaluation Assurance Level 4 (EAL4) as defined in the Common Criteria [CCMB-2009-07-001], [CCMB-2009-07-002], [CCMB-2009-07-004] and augmented with ALC_FLR.1 and AVA_VAN.4. They are all drawn from the Common Criteria V3.1 R2 part 3.

The security assurance components are listed in Table 3. The security assurance requirements defined in Table 4 are defined in section 6.2 of the PP [TPM1.2 PP rev116].

**Table 2: Assurance components**

| # | Assurance class | Assurance Component | Assurance component description |
|---|---|---|---|
| 1 | ADV: Development | ADV_FSP.4 | Complete functional specification |
| 2 | | ADV_ARC.1 | Security architecture description |
| 3 | | ADV_TDS.3 | Basic modular design |
| 4 | | ADV_IMP.3 | Implementation representation of the TSF |
| 5 | AGD: Guidance documents | AGD_OPE.1 | Operational user guidance |
| 6 | | AGD_PRE.1 | Preparative procedures |
| 7 | ALC: Life-cycle support | ALC_CMC.4 | Production support and acceptance procedures and automation |
| 8 | | ALC_CMS.4 | Problem tracking coverage |
| 9 | | ALC_DEL.1 | Delivery procedures |
| 10 | | ALC_DVS | Identification of security measures |
| 11 | | ALC_LCD.1 | Develop defined life-cycle model |
| 12 | | ALC_FLR.1 | Basic flow remediation – augmented |
| 13 | | ALC_TAT.1 | Tools and technique |
| 14 | ASE: Security Target Evaluation | ASE_INT.1 | ST Introduction |
| 15 | | ASE_CCL.1 | Conformance claims |
| 16 | | ASE_SPD.1 | Security problem definition |
| 17 | | ASE_OBJ.2 | Security objectives |
| 18 | | ASE_ECD.1 | Extended components definition |
| 19 | | ASE_REQ.2 | Security requirements |
| 20 | | ASE_TSS.1 | TOE summary specification |
| 21 | ATE: Tests | ATE_COV.2 | Analysis of coverage |
| 22 | | ATE_DPT.1 | Testing: basic design |
| 23 | | ATE_FUN.1 | Functional testing |
| 24 | | ATE_IND.2 | Independent testing |
| 25 | AVA: Vulnerability assessment | AVA_VAN.4 | Methodical vulnerability analysis -augmented |

## 7.3 Security Requirements rationale

The security requirements rationale of the TOE are defined and described in the PP [10] section 6.3 and in the following table.

**Table 3: Security requirements rationale**

| | O.Fail_Secure | O.Import |
|---|---|---|
| FCS_COP.1/VAF | X | X |

# 8 TOE SUMMARY SPECIFICATIONS

The following sections describes how the Security Functional Requirements are fulfilled by the TOE.

The concept of security feature is introduced to provide a common coverage of SFRs that are logically linked.

## 8.1 TOE Security features

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

1) SF_CRY: Cryptographic Support

2) SF_I&A: Authentication and Identification

3) SF_ACC: Access Control

4) SF_GEN: General

5) SF_PNB: Protection and Non-Bypassability

6) SF_TST: Test

### *8.1.1 SF_CRY: Cryptographic support*

There are several functions within the TOE related to cryptographic support: generation of random numbers, generation of asymmetric key pairs, RSA digital signature (generation and verification), data encryption and decryption, key destruction, the generation of hash values and the generation and verification of MAC values.

The TOE supports the generation of cryptographic keys in accordance with the specified cryptographic key generation algorithm *RSA key generator* and specified cryptographic key sizes RSA 512, 1024 and 2048 bits that meet the following: P1363 [IEEE P1363-2000]. The source of randomness is the internal random generator (RNG).

The covered security functional requirement is FCS_CKM.1.

The TOE supports the generation of symmetric cryptographic AES keys in accordance with the specified cryptographic key generation algorithm *AES key generator* and specified cryptographic key sizes 128 bits.

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

The TOE supports the destruction of cryptographic keys by erasure of volatile memory areas containing cryptographic keys or overwriting value of key encryption keys.

The covered security functional requirement is FCS_CKM.4.

The TOE performs the hash calculation in accordance with the specified cryptographic algorithm SHA-1 (cryptographic key sizes not available) that meets FIPS PUB 180-2 [FIPS 180-2].

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

The TOE performs HMAC calculation and verification in accordance with the specified cryptographic algorithm HMAC (SHA-1) and cryptographic key sizes 160 bits that meet RFC2104 [RFC2104] and FIPS PUB 180-2 [FIPS 180-2].

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

The TOE performs signature generation and signature verification in accordance with the specified cryptographic algorithm RSA signature scheme [TPM Part1 116] section 31.2.1, 31.2.2 and 31.2.3, and cryptographic key sizes RSA 512, 1024 and 2048 bits that meet PKCS#1 V2.0. The TOE uses the internal RNG as the source for any randomness that the process may require.

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

The TOE performs encryption and decryption in accordance with the specified cryptographic algorithm RSA encryption scheme [TPM Part1 116] section 31.1.1, 31.1.2 and cryptographic key sizes RSA 512, 1024 and 2048 bits that meet PKCS#1v2.0 [PKCS#1].

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

The TOE performs the encryption and decryption in accordance with the specified cryptographic algorithm AES in the CTR mode as described in [TPM Part1 116] section 31.1.3 and cryptographic key sizes of 128 bits that meet FIPS PUB 197 [FIPS 197] and the specified cryptographic algorithm MGF1 and cryptographic key sizes of 160 bits that meet PKCS#1v2.0 [PKCS#1].

The covered security functional requirement is FCS_COP.1/SymEnc and FCS_COP.1/SymEnc2.

The TOE provides a true random number generator, consisting of an analogue circuit and a digital postprocessing function, implementing the generation of internal random numbers. The TOE provides random numbers that meet AIS31 Class P2.

The covered security functional requirement is FCS_RNG.1

The SF_CRY "Cryptographic Support" covers the following security functional requirements:

- FCS_CKM.1,
- FCS_CKM.4,
- FCS_COP.1 and
- FCS_RNG.1.

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

The TPM provides four protocols for authentication and identification to authorize the use of entities without revealing the authorization data (AuthData) on the network or the connection to the TPM. The basic premise is to prove knowledge of a shared secret. This shared secret is the identification and authentication data, which is called authorization data in the TPM Main Specification. In both cases, the protocol exchanges nonce-data so that both sides of the transaction can compute a MAC value using shared secrets and nonce-data. Each side generates the MAC value and can compare to the value transmitted. Network listeners cannot directly infer the AuthData from the hashed objects sent over the network.

The first protocol is the "*Object-Independent Authorization Protocol*" (OIAP), which allows the exchange of nonces with a specific TPM. Once an OI-AP session is established, its nonces can be used to authorize the use any entity managed by the TPM. The session can live indefinitely until either party request the session termination. The TPM_OIAP function starts the OIAP session.

The second protocol is the "*Object Specific Authorization Protocol*" (OSAP)". The OSAP allows establishment of an authentication session for a single entity. The session creates non ces that can authorize multiple commands without additional session-establishment overhead, but is bound to a specific entity. The TPM_OSAP command starts the OSAP session. The TPM_OSAP specifies the entity to which the authorization is bound.

The third protocol is the "*Delegation Specific Authorization Protocol*" (DSAP)". The DSAP allows establishment of an authentication session for the delegation model (a delegation of individual TPM owner privileges to individual entities). The session creates nonces that can authorize multiple commands without additional session-establishment overhead, but is bound to a specific entity. The TPM_DSAP command starts the DSAP session.

The TPM provides the transport session protocol. The transport session protocol creates a shared secret and then uses the shared secret to authorize and protect commands sent to the TPM using this session. The protection of the sent command is done by encrypting the sent command using a XOR algorithm with a one-time pad.

The TOE allows access to commands and objects with the “World” access on behalf of the user to be performed before the user is authenticated/identified. Each user has to be successfully authenticated/identified before allowing any other TSF-mediated actions on behalf of that user. The TOE controls the access to all protected functions (e.g. commands) and shielded locations in accordance to the access-rights only through the authentication mechanism, i.e. by supplying the appropriate authentication/identification token (a 20 byte long HMAC value). A re-authentication of users is done by using the authentication protocol with a new *nonce* for each message and response. The access-rights of commands, data and keys are defined by security attributes (see PP [8], Table 1). The TOE authenticates any user's claimed identity and reacts on the detection of unsuccessful authentication attempts occur related to the same user according to the rule “dictionary attack”.

The covered security functional requirements are

- FIA_UID.1
- FIA_UAU.1
- FIA_UAU.4,
- FIA_UAU.5,
- FIA_UAU.6 and
- FIA_AFL.1.

The TOE supports the management of TSF data by restricting the ability to modify and create the authentication data to different roles (e.g. TPM owner, User under physical presence, Entity owner, authorized user) based on different rules and restricting the ability to reset the TPM dictionary attack mitigation mechanism and the creation of migration tickets to the TPM owner, by using access control mechanisms during the command processing.

The covered security functional requirements are: FMT_MTD.1/AuthData, FMT_MTD.1/Deleg, FMT_MTD.1/Lock and FMT_MTD.1/MigK.

The TOE associate user security attributes (e.g. authData, locality, physical presence, authorization handle and shared secret if the subject is a OSAP session and authorization associated with the delegation blob if the subject is a DSAP session) with subjects acting on the behalf of that user. The TOE enforces different rules, implemented in the appropriate command, on the initial association and governing changes of user security attributes with subjects acting on the behalf of users.

The covered security functional requirement is FIA_USB.1.

The SF_I&A “Authentication and Identification” covers the following security functional requirements:

- FIA_UID.1,
- FIA_UAU.1
- FIA_UAU.4,
- FIA_UAU.5,
- FIA_UAU.6,
- FIA_AFL.1,
- FMT_MTD.1 and
- FIA_USB.1.

### 8.1.3 *SF_ACC: Access control*

The TOE provides the security function policies TPM Mode Control SFP (MCT_SFP), Delegation SFP (Del_SFP), Key Management SFP (KeyM_SFP), Key Migration SFP (KMig_SFP), Measurement And Reporting SFP (M&R_SFP), Non-volatile Storage SFP (NVS_FSP), Monotonic Counter SFP (MC-SFP), Export and Import of Data (EID_SFP) and Direct Anonymous Attestation Protocol SFP (DAA_SFP) to protect the sensitive subjects, objects and operations of the TOE. The security policies are described in section 8.2 and in the PP [8], section 6.1.

The covered security functional requirements are:

- FDP_ACC.1/Modes
- FDP_ACC.1/Deleg
- FDP_ACC.1/KeyMan
- FDP_ACC.1/MigK
- FDP_ACC.1/M&R
- FDP_ACC.1/NVS
- FDP_ACC.1/MC
- FDP_ACC.1/EID
- FDP_ACC.1/DAA.

The TOE enforces the different security function policies on subjects (e.g. commands, roles), objects (e.g. keys, user data) and operations (e.g. signature generation, encryption and decryption) based on different security attributes (e.g. TPM_AUTH_DATA_USAGE, TPM_KEY_USAGE, TPM_KEY_FLAGS). Any processing is only allowed if the respective security attribute has the correct value.

The covered security functional requirements are:

- FDP_ACF.1/Modes
- FDP_ACF.1/Deleg
- FDP_ACF.1/KeyMan
- FDP_ACF.1/MigK
- FDP_ACF.1/M&R
- FDP_ACF.1/NVS
- FDP_ACF.1/MC
- FDP_ACF.1/EID
- FDP_ACF.1/DAA.

For the TPM different operational modes are defined by different security attributes. The security attributes are stored in structures at shielded locations. The management of the security attributes (e.g. the ability to modify, set to default value, to delete, to enable, to disable, to create) are restricted to different roles und sometimes additionally based on different rules. These restrictions are defined in different structures and are stored at shielded locations or directly programmed in the specific commands. The TOE checks if there are no restrictions violated before processing the management of the security attribute. This functionality is used in principle for all security functional requirements of FMT_MSA.1.

The covered security functional requirements are:

- FMT_MSA.1/Modes
- FMT_MSA.1/PhysP
- FMT_MSA.1/DFT
- FMT_MSA.1/DT

- FMT_MSA.1/KeyMan
- FMT_MSA.1/KEvi
- FMT_MSA.1/MigK
- FMT_MSA.1/MC
- FMT_MSA.1/DAA

The TOE ensures that only secure values are accepted for security attributes. The covered security functional requirement is FMT_MSA.2. The TOE supports the static security attribute initialization. Different security enforcing policies are allowed to provide permissive and/or restrictive default values for security attributes. The TPM owner and the user “World” under physical presence are allowed to specify alternative initial values to override the default values when an object or information is created. The permissions to change the security attributes are stored in different structures or/and controlled during the command processing. This functionality is used in principle for all security functional requirements of FMT_MSA.3.

The covered security functional requirements are:

- FMT_MSA.3/Deleg
- FMT_MSA.3/KeyMan
- FMT_MSA.3/DAA
- FMT_MSA.3/M&R
- FMT_MSA.3/NVS
- FMT_MSA.3/MC
- FMT_MSA.3/EID.

The TOE ensures that any previous information content of a resource is made unavailable upon the deallocation of the resource from any object by overwriting or deallocation of the specific memory area.

The covered security functional requirement is FDP_RIP.1.

The export and import of user data, outside of the TOE and controlled under the SFP, is done under the control of the Key Management SFP, Key Migration SFP and Export and Import of Data SFP. The TOE enforces the export of the user data with the user data's associated security attributes and ensures that the security attributes are unambiguously associated with the exported user data. The TOE use the security attributes associated with the imported user data and ensures that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

The covered security functional requirements are FDP_ETC.2 and FDP_ITC.2.

The SF_ACC “Access Control” covers the following security functional requirements:

- FDP_ACC.1,
- FDP_ACF.1,
- FDP_ETC.2,
- FDP_ITC.2,
- RDP_RIP.1,
- FMT_MSA.1,
- FMT_MSA.2 and
- FMT_MSA.3.

### 8.1.4 *SR_GEN: General*

The TOE provides the roles: TPM owner, Entity owner, Delegated entity, Entity user, User using operatorAuth and “World” and associates users with roles. The role is bound always on specific authentication token, e.g. for the TPM owner it is the TPM ownership token and for the entity owner it is the entity token. The roles are enforced within the TOE because there are specific commands and specific keys bond to different token.

The covered security functional requirement is FMT_SMR.1.

The TOE performs the following management functions: - Management of the TPM modes of operation, - Management of Delegation Tables and Family Tables, - Management of security attributes of keys, - Management of security attributes of PCR, - Management of security attributes of NV storage areas, - Management of security attributes of monotonic counters and - Reset the Action Flag of TPM dictionary attack mitigation mechanism.

The covered security functional requirement is FMT_SMF.1.

The TOE provides an authentication functionality to consistently interpret authentication reference data of the TPM owner, delegated entities, owner of entities, user of entities and User using operatorAuth, when shared between the TSF and another trusted IT product and uses roles when interpreting the TSF data from another trusted IT product.

The covered security functional requirement is FPT_TDC.1.

The TOE provides the transmission and reception of user data in encrypted manner, to protect the user data from unauthorized disclosure.

The covered security functional requirements are FDP_UCT.1/Exp and FDP_UCT.1/Imp.

The TOE provides the transmission and reception of user data in encrypted and signed manner, to protect the user data from undiscovered modification, deletion, insertion and replay errors (only required for sessions). Interpreting the signature and the decrypted user data command input the TOE is able to determine, whether modification, deletion and insertion and replay has occurred.

The covered security functional requirements are FDP_UIT.1/Data and FDP_UIT.1/Session.

The TOE provides the generation of an audit record of the event Transport session including different information (e.g. type and outcome of event).

The covered security functional requirement is FAU_GEN.1.

The TOE provides reliable time stamps as number of ticks since start of the tick session.

The covered security functional requirement is FPT_STM.1.

The TOE provides the generation of evidence of origin for transmitted data at the request of the originator and is able to verify the evidence of origin of transmitted data to recipient, by calculation and verifying a digital signature of the data.

The covered security functional requirements are FCO_NRO.1/STS and FCO_NRO.1/M&R.

The SF_GEN “General” covers the following security functional requirements:

- FMT_SMR.1,
- FMT_SMF.1,
- FDP_TDC.1,
- FDP_UCT.1,
- FDP_UIT.1,
- FPT_STM.1,
- FCO_NRO.1 and
- FAU_GEN.1.

### 8.1.5 *SF_PNB: Protection and non-bypassability*

The field upgrade functionality guarantees the verification of the authenticity of the firmware file during the loading process in the memory of the TOE. The integrity of the TOE firmware and the non-bypassability of the verification are covered by the security functional requirement FCS_COP.1/VAF.

The TOE preserves a secure state when a 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 (including AES encryption/ decryption), authorization and failure of operating conditions (power supply) occurs. The Neslib library provides protection against fault attacks (FA) on those cryptographic functions thanks to code signature mechanism. The TOE maintains also a secure state after power failure thanks to an atomicity engine that guarantees non volatile memory integrity and coherence when power failure happens during writing operations.

The covered security functional requirement is FPT_FLS.1.

The TOE supports the Direct Anonymous Attestation Protocol.

The covered security functional requirement is FPR_UNL.1.

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

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

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

### 8.1.6 *SF_TST: Test*

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

The covered security functional requirement is FPT_TST.1.

### 8.1.7 Security Functional Requirements coverage by security features

**Table 4: Summary table of SFR coverage by security features**

| SFR | SF_CRY | SF_I&A | SF_ACC | SF_GEN | SF_PNB | SF_TST |
|---|---|---|---|---|---|---|
| FMT_SMR.1 | | | | X | | |
| FMT_SMF.1 | | | | X | | |
| FDP_ACC.1/Modes | | | X | | | |
| FDP_ACC.1/Deleg | | | X | | | |
| FDP_ACC.1/KeyMan | | | X | | | |
| FDP_ACC.1/MigK | | | X | | | |
| FDP_ACC.1/M&R | | | X | | | |
| FDP_ACC.1/NVS | | | X | | | |
| FDP_ACC.1/EID | | | X | | | |
| FDP_ACC.1/MC | | | X | | | |
| FDP_ACC.1/DAA | | | X | | | |
| FDP_ACF.1/Modes | | | X | | | |
| FDP_ACF.1/Deleg | | | X | | | |
| FDP_ACF.1/KeyMan | | | X | | | |
| FDP_ACF.1/MigK | | | X | | | |
| FDP_ACF.1/M&R | | | X | | | |
| FDP_ACF.1/NVS | | | X | | | |
| FDP_ACF.1/MC | | | X | | | |
| FDP_ACF.1/EID | | | X | | | |
| FDP_ACF.1/DAA | | | X | | | |
| FMT_MSA.1/Modes | | | X | | | |
| FMT_MSA.1/PhysP | | | X | | | |
| FMT_MSA.1/DFT | | | X | | | |
| FMT_MSA.1/DT | | | X | | | |
| FMT_MSA.1/KeyMan | | | X | | | |
| FMT_MSA.1/MigK | | | X | | | |
| FMT_MSA.1/Kevi | | | X | | | |
| FMT_MSA.1/MC | | | X | | | |
| FMT_MSA.1/DAA | | | X | | | |
| FMT_MSA.2 | | | X | | | |
| FMT_MSA.3/Deleg | | | X | | | |
| FMT_MSA.3/KeyMan | | | X | | | |
| FMT_MSA.3/M&R | | | X | | | |
| FMT_MSA.3/NVS | | | X | | | |
| FMT_MSA.3/MC | | | X | | | |
| FMT_MSA.3/EID | | | X | | | |
| FMT_MSA.3/DAA | | | X | | | |
| FDP_ETC.2 | | | X | | | |

| SFR | SF_CRY | SF_I&A | SF_ACC | SF_GEN | SF_PNB | SF_TST |
|---|---|---|---|---|---|---|
| FDP_ITC.2 | | | X | | | |
| FDP_RIP.1 | | | X | | | |
| FCS_CKM.1 | X | | | | | |
| FCS_CKM.1/AES | X | | | | | |
| FCS_CKM.4 | X | | | | | |
| FCS_RNG.1 | X | | | | | |
| FCS_COP.1/SHA | X | | | | | |
| FCS_COP.1/HMAC | X | | | | | |
| FCS_COP.1/RSA_Sig | X | | | | | |
| FCS_COP.1/VAF | | | | | | X |
| FCS_COP.1/RSA_Enc | X | | | | | |
| FCS_COP.1/SymEnc | X | | | | | |
| FCS_COP.1/SymEnc2 | X | | | | | |
| FMT_MTD.1/AuthData | | X | | | | |
| FMT_MTD.1/Deleg | | X | | | | |
| FMT_MTD.1/Lock | | X | | | | |
| FMT_MTD.1/MigK | | X | | | | |
| FIA_UID.1 | | X | | | | |
| FIA_UAU.1 | | X | | | | |
| FIA_UAU.4 | | X | | | | |
| FIA_UAU.5 | | X | | | | |
| FIA_UAU.6 | | X | | | | |
| FIA_AFL.1 | | X | | | | |
| FIA_USB.1 | | X | | | | |
| FPT_TDC.1 | | | | X | | |
| FCO_NRO.1/M&R | | | | X | | |
| FCO_NRO.1/STS | | | | X | | |
| FDP_UCT.1/Exp | | | | X | | |
| FDP_UCT.1/Imp | | | | X | | |
| FDP_UIT.1/Data | | | | X | | |
| FDP_UIT.1/Sesson | | | | X | | |
| FAU_GEN.1 | | | | X | | |
| FPT_STM.1 | | | | X | | |
| FPT_FLS.1 | | | | | X | |
| FPR_UNL.1 | | | | | X | |
| FPT_PHP.3 | | | | | X | |
| FPT_TST.1 | | | | | | X |

## Appendix A ABBREVIATIONS

### A.1 Abbreviations

Table 5: Abbreviations

| Term | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| CC | Common Criteria |
| DES | Data Encryption Standard |
| DSAP | Delegate Specific Authorization Protocol |
| EK | Endorsement Key |
| FIPS | Federal Information Processing Standard |
| GPIO | General Purpose I/O |
| HMAC | Keyed-Hashing for Message Authentication |
| LPC | Low Pin Counter (Intel protocol) |
| NIST | National Institute of Standards and Technology |
| NV | Non-volatile (memory) |
| OIAP | Object-Independent Authorization Protocol |
| OSAP | Object Specific Authorization Protocol |
| PCR | Platform Configuration Register |
| RSA | Rivest Shamir Adelman |
| RTM | Root of Trust for Measurement |
| RTR | Root of Trust for Reporting |
| SAR | Security Assurance Requirement |
| SFR | Security Functional Requirement |
| SHA | Secure Hash Algorithm |
| SPI | Serial Peripheral Interface |
| SRK | Storage Root Key |
| T=0, T=1 | IC Card communication protocol |
| TCG | Trusted Computed Group |
| TPM | Trusted Platform Module |
| TPME | TPM Manufacturer |
| TSS | TPM Software Stack |

## Appendix B REFERENCED DOCUMENTS

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

**[1]** [CCMB-2009-07-001]

Common Criteria for Information Technology Security Evaluation, Version 3.1, Part 1: Introduction and general model, Revision 3, July 2009

**[2]** [CCMB-2009-07-002]

Common Criteria for Information Technology Security Evaluation, Version 3.1, Part 2: Security functional components, Revision 3, July 2009

**[3]** [CCMB-2009-07-004]

Common Criteria for Information Technology Security Evaluation, Version 3.1, Part 3: Security assurance components, Revision 3, July 2009

**[4]** [CCMB-2009-07-04]

Common Methodology for Information Technology Security Evaluation (CEM) Evaluation Methodology, Version 3.1, Rev 3, July 2009

**[5]** [TPM Part1 116]

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

**[6]** [TPM Part2 116]

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

**[7]** [TPM Part3 116]

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

**[8]** [PC Client TIS 1.21]

TCG PC Client Specific TPM Interface Specification (TIS) for TPM Family 1.2 Version 1.21 Final – Revision 1.00 – May 2011

**[9]** [TCG Glossary]

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

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

Protection Profile PC Client Specific TPM, Family 1.2 Level 2 Revision 116 (Version 1.2), 18 May 2011, TCG

**[11]** [IEEE P1363-2000]

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

**[12]** [FIPS 180-2]

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

**[13]** [RFC2104]

RFC2104 - HMAC: Keyed-Hashing for Message Authentication

**[14]** [PKCS#1]

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

**[15]** [FIPS 197]

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

**[16]** [FIPS 46-3]

FIPS Publication, Data Encryption Standard (DES) reaffirmed 1999 October 25

**[17]** [SP 800-17]

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

**[18]** [AIS 31]

Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren, Version 1, 25 Sept 2001

**[19]** [ST33ZP24PVSC DS]

ST33TPM12LPC Datasheet V6, STMicroelectronics, 2012

**[20]** [ST33ZP24PVSH DS]

ST33TPM12LPC Datasheet V7, STMicroelectronics, 2012

**[21]** [ST33ZP24PVSP DS]

ST33TPM12LPC Datasheet V10, STMicroelectronics, 2012

**[22]** [FU_12D8 RN]

Field upgradable firmware 1.2.D.8 for ST33TPM12LPC Trusted Platform Module, STMicroelectronics, 2012

**[23]** [FU_12DC RN]

Field upgradable firmware 1.2.D.C for ST33TPM12LPC Trusted Platform Module, STMicroelectronics, 2012

**Please Read Carefully:**

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

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

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

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

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

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

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

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

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

**www.st.com**