Protection Profile
PC Client Specific TPM

TPM Library specification Family "2.0"
Level 0 Revision 1.38
30 January 2018
Version 1.1

Contact: admin@trustedcomputinggroup.org

- **Work in Progress:**

This document is an intermediate draft for comment only and is subject to change without notice. Readers should not design products based on this document.

TCG

TCG Public Review

Disclaimers, Notices, and License Terms

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Without limitation, TCG disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and TCG disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

This document is copyrighted by Trusted Computing Group (TCG), and no license, express or implied, is granted herein other than as follows: You may not copy or reproduce the document or distribute it to others without written permission from TCG, except that you may freely do so for the purposes of (a) examining or implementing TCG specifications or (b) developing, testing, or promoting information technology standards and best practices, so long as you distribute the document with these disclaimers, notices, and license terms.

Contact the Trusted Computing Group at www.trustedcomputinggroup.org for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

Table of Contents

Tables

This page is intentionally left blank.

1 **Version History**

Version	Date	Description
1.0	10.12.2014	First official release
1.1	30.01.2018	Update to TPM Library 2.0 level 0 revision 1.38 Update to PC Client Platform Profile 1.03 Update to Common Criteria 3.1 R5

2

1. Scope

This protection profile describes the security requirements for the Trusted Computing Group (TCG) PC Client Specific Trusted Platform Module (TPM) Family 2.0; Level 0 conforming to the Common Criteria version 3.1 revision 5.

A TPM designer MUST be aware that for a complete definition of all requirements necessary to build a TPM, the designer MUST use the Trusted Computing Group TPM Library specification and the PC client specific specification for all TPM requirements. Security targets for Common Criteria evaluation of PC Client Specific Trusted Platform Module MUST be strictly conformant to this protection profile.

1.1 Key words

The key words “MUST,” “must”, “MUST NOT,” “must not”, “REQUIRED,” “required”, “SHALL,” “shall”, “SHALL NOT,” “shall not”, “RECOMMENDED,” “recommended”, “MAY,” “may”, “OPTIONAL”, and “optional” in this document normative statements are used as described in RFC-2119. “SHOULD”, “should”, “SHOULD NOT”, and “should not” have an additional meaning and are to be interpreted as described in Common Criteria Part 1, p. 11.

1.2 Statement Type

Please note a very important distinction between different sections of text throughout this document. There are two distinctive kinds of text: application notes as informative comment and normative statements. Because most of the text in this protection profile will are normative statements, the authors have informally defined it as the default and, as such, have specifically called out text which is informative comment. This means that unless text is specifically marked as informative comment, it is considered to be normative statements.

2. PP Introduction

2.1 PP Reference

Title: Protection Profile PC Client Specific Trusted Platform Module Specification Family 2.0; Level 0; Revision 1.38 Version 1.1 (PP PCCS TPM F2.0 L0 r1.38 V1.1)

Sponsor: Trusted Computing Group

CC Version: 3.1 (Release 5)

Assurance level: EAL 4 augmented with ALC_FLR.1 and AVA_VAN.4

Document version: 1.1

Keywords: trusted computing group, trusted platform module, PC client specific TPM

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 a device that implements the functions defined in the TCG Trusted Platform Module Library Specification, version 2.0, [7], [8], [9], [10] and the PC client specific interface specification [11]. The TCG Trusted Platform Module Library specification describes the design principles, the TPM structures, the TPM commands and supporting routines for the commands. The TPM PC client specific interface specification describes the additional features that must be implemented by a TPM for a PC Client platform.

The TOE consists of

(1) TPM hardware,

(2) TPM firmware,

(3) TPM guidance documentation.

The TPM hardware is typically implemented as a single-chip component that is attached to the PC platform using a low-performance interface. It has processor, RAM, ROM and Flash memory and may have special components to support random number generation and cryptographic operations. The TPM firmware is running on the TPM platform. The TPM guidance documentation provides the necessary information for secure usage of the TOE by customers and users.

2.2.2 TOE Usage and Security Features

The TPM library specification describes the TPM protections in terms of Protected Capabilities and Protected Objects (cf. [7], chapter 10 for details). A Protected Capability is an operation that must be correctly performed for a TPM to be trusted and therefore is in the scope of the CC evaluation as part of the TOE security functionality (TSF). A Protected Object is data that must be protected for a TPM operation to be trusted. The TSF performs all operations with Protected Objects inside the TPM. The TSF protects the confidentiality of Protected Objects when exported from the TPM and checks the integrity of Protected objects

when imported into the TPM. The TOE provides physical protection for Protected Objects residing in the TPM.

The TPM provides methods for collecting and reporting identities of hardware and software components of a computer system platform. The computer system report is generated by a Trusted Computing Base (TCB) that includes the TPM and allows determination of expected behavior. From the report provided by the TCB, there is trust in the computer system platform.

There are commonly three Roots of Trust in a trusted platform; a root of trust for measurement (RTM), root of trust for reporting (RTR) and root of trust for storage (RTS). In TCG systems roots of trust are components that must be trusted because misbehavior cannot be detected. The RTM is a computing engine capable of making inherently reliable integrity measurements and maintaining an accurate summary of values of integrity digests and the sequence of digests. The RTR is a computing engine capable of reliably reporting information held by the RTM. The RTS provides secure storage for a practically unlimited number of private keys or other data by means of exporting and importing encrypted data.

Support for the Root of Trust for Measurement

The TPM supports the integrity measurement of the trusted platform by calculation and reporting of measurement digests of measured values. Typically the RTM is controlled by the Core Root of Trust for Measurement (CRTM) as the starting point of the measurement. The measurement values are representations of embedded data or program code scanned and provided to the TPM by the measurement agent. The TPM supports cryptographic hashing of measured values and calculates the measurement digest by extending the value of a PCR with a calculated or provided hash value. The PCRs are shielded locations of the TPM which can be reset by TPM reset or a trusted process, and written only through measurement digest extensions, and read.

Root of Trust for Reporting

The TPM holds the Endorsement Primary Seed (EPS) and generates Endorsement Keys (EK) from the EPS. The EK and the corresponding Endorsement Certificates define the trusted platform identities for the RTR. The TPM may be shipped with an EK and a Certificate of the Authenticity of this EK. The EK is bound to the Platform via a Platform Certificate, providing assurance from the certification body of the physical binding and connection through a trusted path between the platform (the RTM) and the genuine TPM (the RTR). The attestation of the EK and the Platform Certificates build the base for attestation of other keys and measurements (cf. [7] chapter 9.5 for details).

Root of Trust for Storage

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

TPM. The TPM uses symmetric cryptographic algorithms to encrypt data and keys and may implement asymmetric cryptographic algorithms of equivalent strength.

Platform Key Hierarchy

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

Other Security Services and Features

The TOE provides cryptographic services for hashing, asymmetric encryption and decryption, asymmetric signing and signature verification, symmetric encryption and decryption, symmetric signing and signature verification and key generation. The Hash function SHA-1 and SHA-256 are provided as a cryptographic service to external entities for measurements and used internally for user authentication, signing and key derivation. A TOE is required to implement asymmetric algorithms: where the current specification supports RSA with 2048 bits for digital signature, secret sharing and encryption and ECC algorithms with P-256 and BN-256 curves for digital signatures and secret sharing. The TOE provides symmetric encryption and decryption of AES-128 in CFB mode and perhaps additional algorithms in CBC, CTR, ECB (not recommended) and OFB modes of operation. The TOE implements symmetric signing and signature verification by means of HMAC described in [16]. The TOE generates three types of keys: Ordinary keys are generated using the random number generator to seed the key computation. Primary Keys are derived from a Primary Seed and key parameters by means of a key derivation function. Derived Keys are derived from the sensitive value of the parent and key parameters by means of a key derivation function.

The TPM stores persistent state associated with the TPM in NV memory and provides NV memory as a shielded location for data of external entities. The platform and entities authorised by the TPM owner control allocation and use of the provided NV memory. The access control may include the need for authentication of the user, delegations, PCR values and other controls.

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

2.2.3 Non-TOE Hardware, Firmware and Software

The TPM is a hardware component of a computer system. The Platform firmware and the operating system of this computer system interact with the TPM by sending commands to the TPM and receiving responses of the TPM through the interface described in [7] and [0]. Further, the TPM is able to obtain the indications _TPM_Init, locality and an optional feature physical presence via its I/O interface, and adjust its internal state accordingly. Therefore the TOE is a passive device controlled by the software running on the computer system.

2.2.4 TPM Life Cycle

The TPM life cycle may be described in four phases: Development, Manufacturing, Platform Integration and Operational usage. Because the PC client specific TPM supports Field Upgrade the TPM life cycle distinguishes two cases.

- Case 1: The TPM hardware and firmware are manufactured and delivered together.
- Case 2: The TOE firmware component is installed (as a replacement or an augmentation of the previously loaded TPM firmware) after delivery of the TOE hardware component to the platform vendor or the end user.

The full Field Upgrade (cf. [7], chapter 12.5.2) does change the TOE and the incremental Field Upgrade may change the TOE. The TPM life cycle is also linked to the life cycles of the EPS, PPS and SPS and their corresponding key hierarchies.

Case 1 of the TPM life cycle can be summarised as follows.

- Development of the TPM (Phase 1)

  The Development of the TPM (Phase 1) comprises the development of the TPM hardware and the TPM firmware.

- Manufacturing and Delivery of the TPM (Phase 2)

  The Manufacturing Phase comprises the production of the integrated circuit implementing complete or parts of TPM firmware, the loading of TPM firmware parts stored in EEPROM or Flash memory, testing and delivery to the platform vendor.

  In this phase the TPM manufacturer may inject EPS and PPS but whenever the TPM is powered on and no EPS, PPS or SPS is present the missing primary seeds will be generated automatically and may be changed afterwards. The TPM manufacturer may generate an EK and the corresponding Endorsement Certificate as evidence for its genuine TPM.

  This phase ends with TPM delivery to the customer.

- Platform Integration (Phase 3)

  The TPM is installed in the platform, equipped with TPM and platform specific keys and certificates, and delivered to the customer of the platform.

  In this phase the platform vendor may equip the TPM with the PPS, Platform Primary Key, Platform Keys and corresponding Platform Certificates. The Platform hierarchy and the Endorsement hierarchy (based on the EPS created by the TPM manufacturer or the Platform manufacturer) may be bound by cross certification.

- Operational usage (Phase 4)

  In the Operational Phase the TPM is prepared for operational usage and used in the environment of the end user. The preparative procedures for operational usage include secure acceptance of the delivered TOE, taking and releasing ownership and establishing the Storage key hierarchy for protection of owner-related and other User data and TSF data of the TPM outside the TPM.

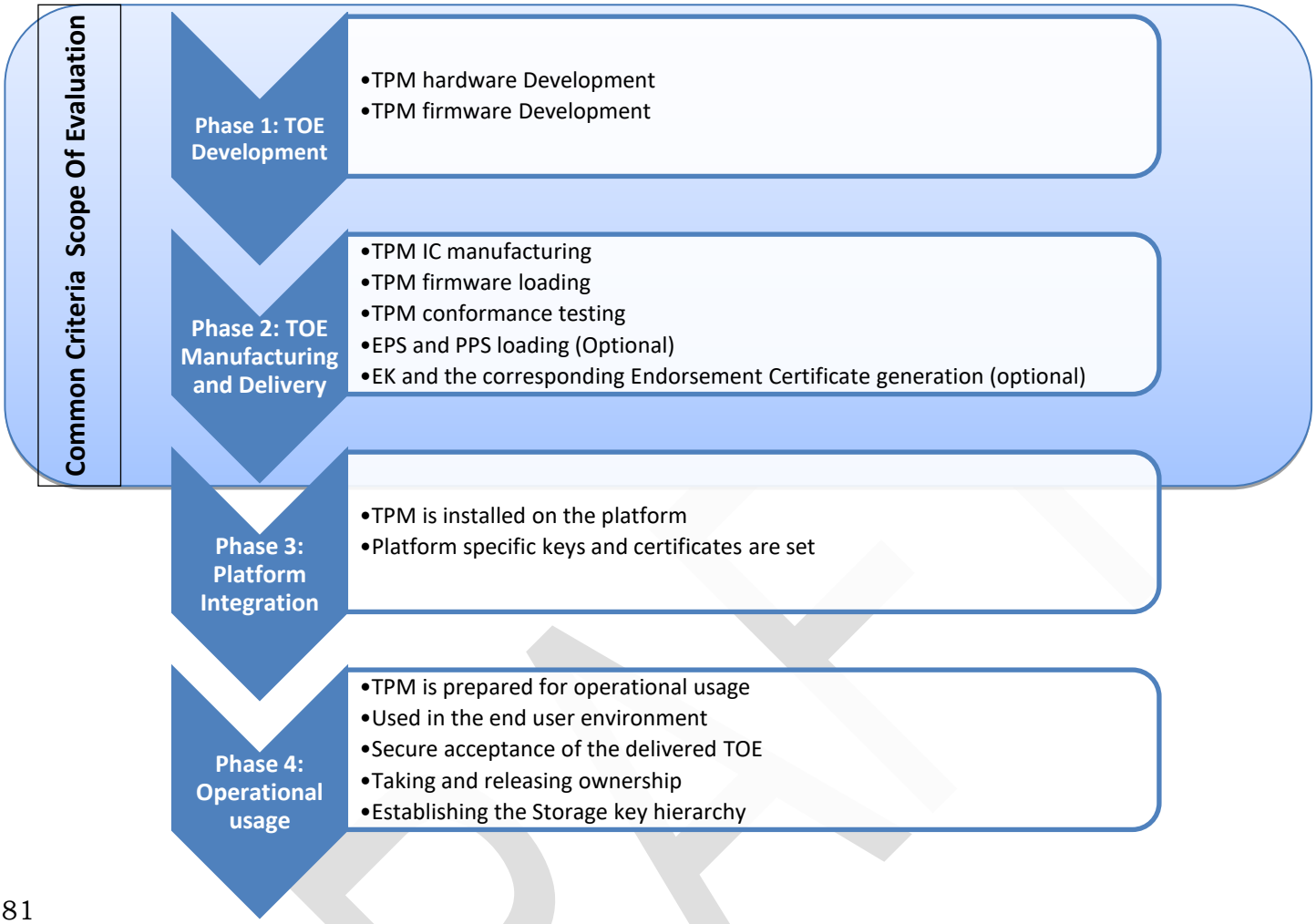

Figure 1: TPM Life Cycle case 1

In case 2 of the TPM life cycle the TPM hardware and parts of the TPM firmware of a previously certified TPM are used for access, integrity and authenticity control of the installation of the new firmware running on the same hardware and building a new TPM. The parts of the previously certified TPM may be run through the life cycle as in case 1 or in case 2.

The following steps describe the life cycle case 2 for the upgraded firmware parts only. The TOE hardware is as already delivered to the platform vendor or the end user.

- Development of the TPM (Phase 1)

  The Development of the TPM (Phase 1) comprises the development and testing of the TPM firmware upgrades to be installed on hardware of a previous TPM.

- Manufacturing of the TPM (Phase 2)

The TPM manufacturer delivers the firmware upgrade for Field Upgrade to the platform vendor as their customer.[1]

- Platform Integration (Phase 3)

  The platform vendor uses the Field Upgrade functionality[2] to install the new TPM firmware on hardware of a previous TPM before delivery of the platform to the end user.

  Note the platform vendor may use different ways for delivery of the firmware upgrade to the end user, e.g. using update mechanisms of operating systems running on the platform.

- Operational usage (Phase 4)

  The platform vendor or the end user may use the Field Upgrade functionality to install the new TPM firmware on hardware of a previous TPM after delivery of the platform to the end user. The preparative procedures for operational usage of the new certified TPM include secure acceptance procedures for use by the end user.

  The Field Upgrade preserves User data (NV index allocations and contents, persistent object allocations and contents) and TSF data (EPS, PPS and SPS, hierarchy authentication reference data *authValue*, *authPolicy* and *proof*, *lockoutAuth, lockoutPolicy,* lockout parameters, the PCR *authValue* and *authPolicy* values, *clock*). Thus the Field Upgrade does not change the Storage key hierarchy for protection of owner-related and other User data and TSF data. After Field Upgrade the new TPM will be ready for operational use in the environment of the end user.

The installation of the new firmware may be performed in Phase 3 or Phase 4. The previous TOE requires authorisation for firmware upgrade and verifies the integrity and authenticity of TPM firmware upgrade data as provided by the TPM firmware manufacturer. But the new TPM may or may not be a certified TPM depending on the TPM vendor or platform vendor certification policy. Thus the user of the TPM shall be made aware of these changes, whether the installed firmware is certified, and which version of a certified TPM is installed.

[1] The TPM manufacturer may use the field upgrade process as well but this is not expected as the TPM vendor may use manufacturing utilities.

[2] The field upgrade implementation may be proprietary or compliant to the library specifications but must fulfill the SFRs defined in this protection profile.

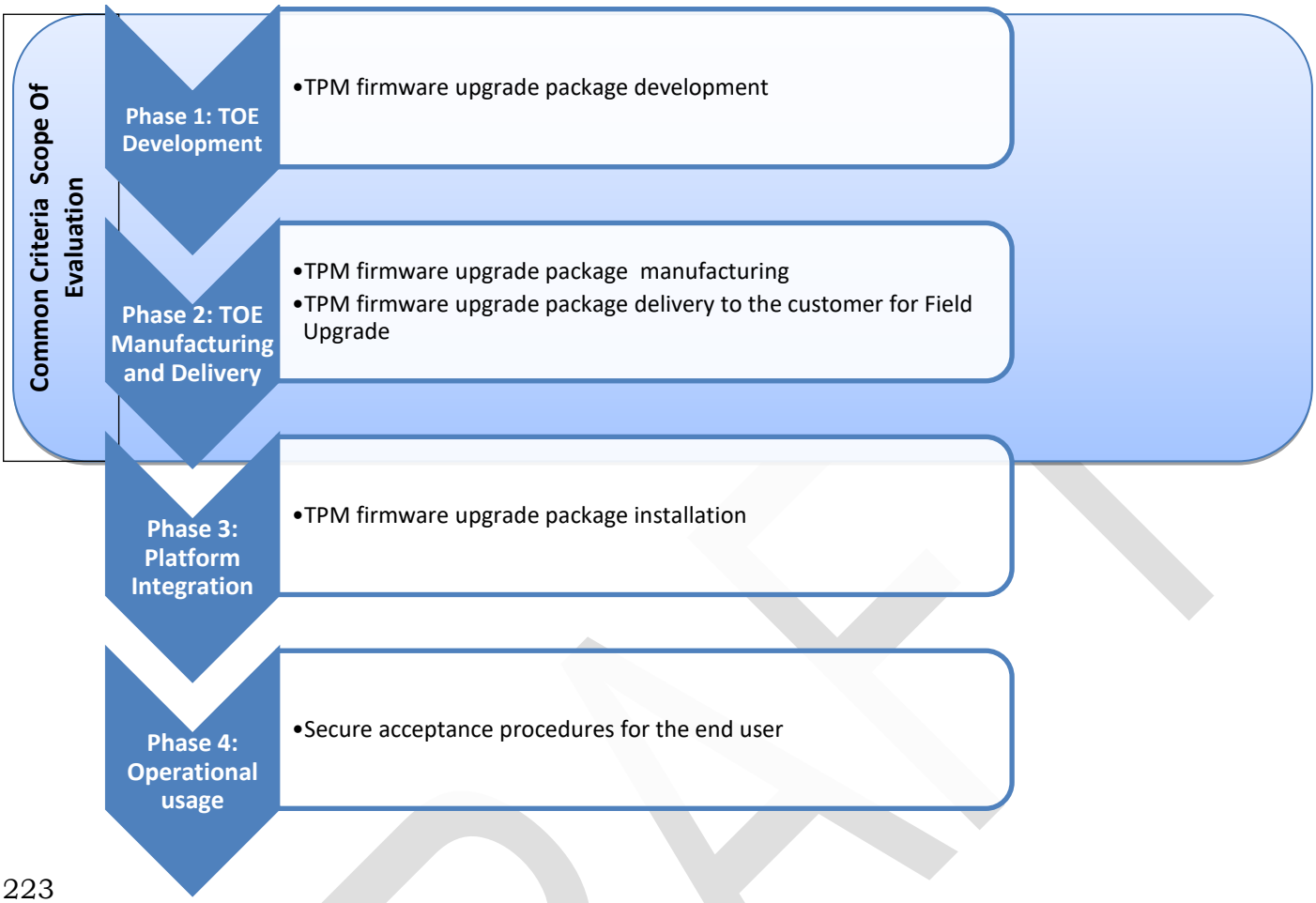

Figure 2 TPM Life Cycle case 2

The Common Criteria evaluation covers the Development of the TOE (Phase 1), the Manufacturing of the TPM (phase 2) up to the delivery to the platform vendor under the development environment (cf. CC part 1 [1], paragraph 157) in the evaluator activity of class ALC: Life-cycle support. The concrete state of the TPM when delivered to the platform vendor as customer of the TPM vendor depends on the vendor configuration options. A TPM can be delivered with no key, or with an Endorsement Key, or with an Endorsement Key and Endorsement Certificate, or with a Platform Key and Platform Certificate. The security target shall describe all configurations of the TOE as delivered to the platform vendor. Details on these configurations will be provided for evaluator activities of families ALC_CMS and ALC_DEL. The user guidance provide by the TPM vendor shall describe the requirement and general procedures and the supplier of the certified TOE shall obey these procedures enabling the end users' acceptance of the certified version and configuration of the delivered TOE. (cf. element AGD_PRE.1.1C for details).

3. Conformance Claims

The following sections describe the conformance claims of the Protection Profile PC Client Specific Trusted Platform Module.

3.1 CC Conformance Claim

This Protection Profile claims to be conformant with the Common Criteria version 3.1 Release 5 as follows

- Part 2 extended,
- Part 3 conformant.

3.2 Conformance with Packages

This PP is conformant to assurance package EAL4 augmented with ALC_FLR.1 and AVA_VAN.4 defined in CC part 3.

3.3 Conformance with other Protection Profiles

This PP does not claim conformance to any other PP.

3.4 Conformance Statement

This PP requires **strict** conformance of any ST or PP, that claims conformance to this PP.

4. Security Problem Definition

The following sections describe the security problem definition of the Protection Profile PC Client Specific Trusted Platform Module.

4.1 Assets

This section of the security problem definition shows the assets of the TOE to be protected and the threats that are considered.

The assets are:

- Protected Objects, operations, security attributes and authorisation data as defined in Table 8.
- Objects, operations and security attributes for the TPM state control SFP as defined in Table 9.

4.2 Threats

This section of the security problem definition shows the threats that are to be countered by the TOE, its development environment, its operational environment, or a combination of these three. A threat consists of a threat agent, an asset (either in the operational or in the development environment) and an adverse action of that threat agent on that asset.

Table 1: Threats

#	Threat	Description
1	T.Compromise	An undetected compromise of the data in shielded locations may occur as a result of an attacker (whether an insider or outsider) attempting to perform actions that the individual or capability is not authorised to perform.
2	T.Bypass	An unauthorised individual or user may tamper with TSF, security attributes or other data in order to bypass TOE security functions and gain unauthorised access to TOE assets.
3	T.Export	A user or an attacker may export data from shielded locations without security attributes or with insecure security attributes, causing the data exported to be erroneous and unusable, to allow erroneous data to be added or substituted for the original data, and/or to reveal secrets.
4	T.Hack_Crypto	Cryptographic key generation or operation may be incorrectly implemented, allowing an unauthorised individual or user to compromise keys generated within the TPM or encrypted data or to modify data undetected.
5	T.Hack_Physical	An unauthorised individual or user of the TOE may cause unauthorised disclosure or modification of TOE assets by physically interacting with the TOE. The attacker may be a

#	Threat	Description
		hostile user of the TOE.
6	T.Imperson	An unauthorised individual may impersonate an authorised user of the TOE (e.g. by dictionary attacks to guess the authorisation data) and thereby gain access to TOE data in shielded locations and protected capabilities.
7	T.Import	A user or attacker may import data without security attributes or with erroneous security attributes, causing key ownership and authorisation to be uncertain or erroneous and the system to malfunction or operate in an insecure manner.
8	T.Insecure_State	The TOE may start-up in an insecure state or enter an insecure state, allowing an attacker to obtain sensitive data or compromise the system.
9	T.Intercept	An attacker may intercept the communication between a user and the TPM subjects to gain knowledge of the commands and data sent to the subject or manipulate the communication.
10	T.Malfunction	TOE assets may be modified or disclosed to an unauthorised individual or user of the TOE, through malfunction of the TOE.
11	T.Modify	An attacker may modify data in shielded locations or their security attributes in order to gain access to the TOE and its assets.
12	T.Object_Attr_Change	A user or attacker may create an object with no security attributes or make unauthorised changes to security attribute values for an object to enable attacks.
13	T.Replay	An unauthorised individual may gain access to the system and sensitive data through a “replay” or “man-in-the-middle” attack that allows the individual to capture identification and authentication data.
14	T.Repudiate_Transact	An originator of data may deny originating the data to avoid accountability.
15	T.Residual_Info	A user may obtain information that the user is not authorised to have when the data in shielded locations is no longer actively managed by the TOE (“data scavenging”).
16	T.Leak	An attacker may exploit information which is leaked from the TOE during usage of the TSF in order to disclose confidential assets.

272

4.3 Organisational Security Policies

This section of the security problem definition shows the Organisational Security Policies (OSPs) that are to be enforced by the TOE, its development environment, its operational environment, or a combination of these three. OSPs are rules, practices, or guidelines. These may be laid down by the organisation controlling the operational environment of the TOE, or they may stem from legislative or regulatory bodies. OSPs can apply to the TOE, the operational environment of the TOE, and/or the development environment of the TOE.

Table 2: Organisational Security Policies

#	OSP	Description
1	OSP.Context_Mana-gement	A resource manager shall be able to secure caching of resources without knowledge or assistance from the application that loaded the resource.
2	OSP.Policy_Authorisation	The TPM supports multiple trusted processes obeying the principle of least privilege by means of role based administration and separation of duty by configuring policy authorisation to allow individual entities (trusted processes, specific privileges, operations).
3	OSP.Locality	The TCG platform supports multiple transitive trust chains by means of a mechanism known as locality. The Host Platform's trusted processes assert their locality to the TPM. The TPM guards access to resources, PCRs and NV Storage Space, to keys and data to be imported, and to defined commands depending on the execution environment's privilege level.
4	OSP.RT_Measurement	The root of trust for measurement calculates and stores the measurement digests as hash values of a representation of embedded data or program code (measured values) for reporting.
5	OSP.RT_Reporting	The root of trust for reporting reports on the contents of the RTS. A RTR report is typically a digitally signed digest of the contents of selected values within a TPM (measurement, key properties or audit digest). The authenticity of the assets reported is based on the verification of the signature and the certificate of the signing key.
6	OSP.RT_Storage	The root of trust for storage protects the assets (listed in Table 8 and Table 9) entrusted to the TPM in confidentiality and integrity.
7	OSP.FieldUpgrade	The Platform software is allowed to perform Field Upgrade within the certified TPM or installing a new certified TPM before and after delivery to the end user. The end user shall be aware of the certification and the version of the TPM.
8	OSP.ECDAA	The ECDAA issuer and the TPM owner establish a procedure for attestation without revealing the attestation information

#	OSP	Description
		(i.e. the identity of the TPM).

281

282

283 4.4 Assumptions

284 This section of the security problem definition shows the assumptions that the TOE makes
285 on its operational environment in order to be able to provide security functionality. If the
286 TOE is placed in an operational environment that does not meet these assumptions, the
287 TOE may not be able to provide all of its security functionality anymore.

288 **Table 3: Assumptions to the IT Environment**

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

289

5. Security Objectives

5.1 Security Objectives for the TOE

The security objectives are a concise and abstract statement of the intended solution to the problem defined by the security problem definition. The TOE provides security functionality to solve a certain part of the problem defined by the security problem definition. This part wise solution is called the security objectives for the TOE and consists of a set of statements describing the security goals that the TOE should achieve in order to solve its part of the problem.

Table 4: Security Objectives for the TOE

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

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

#	Objective	Description
		vendor.
22	O.ECDAA	The TPM must support the TPM owner for attestation to the authenticity of measurement digests without revealing the attestation information by implementation of the TPM part of the ECDAA.

5.2 Security Objectives for the Operational Environment

The following table defines the security objectives for the operational environment of the TOE.

Table 5: Security Objectives for the Operational Environment

#	Objective Name	Objective Description
1	OE.Configuration	The TOE must be installed and configured properly for starting up the TOE in a secure state. The security attributes of subjects and objects shall be managed securely by the authorised user.
2	OE.Locality	The developer of the host platform must ensure that trusted processes indicate their correct locality to the TPM and untrusted processes are only able to assert the locality 0 to the TPM.
3	OE.Credential	The IT environment must create EK and AK credentials by trustworthy procedures for the root of trust for reporting.
4	OE.Measurement	The platform part of the root of trust for measurement provides a representation of embedded data or program code (measured values) to the TPM for measurement.
5	OE.FieldUpgradeInfo	The developer via AGD documentation will instruct the admin how to do the upgrade and that the admin should inform the end user regarding the Field Upgrade process, its result, whether the installed firmware is certified or not, and the version of the certified TPM.
6	OE.ECDAA	The ECDAA issuer must support a procedure for attestation without revealing the attestation information based on the ECDAA signing operation.

5.3 Security Objective Rationale

The following table provides an overview of the mapping between the security objective for the TOE and the functional security requirements. The table shows and the rationale demonstrates that each security objective for the TOE is traced back to threats countered by that security objective and OSPs enforced by that security objective; each security objective for the operational environment is traced back to threats countered by that

311 security objective, to OSPs enforced by that security objective, and to assumptions upheld
312 by that security objective. All security objectives counter all threats, enforce all
313 organisational security policies and uphold all assumptions.

314 **Table 6: Security Objective Rationale**

	O.Context_Management	O.Crypto_Key_Man	O.ECDAA	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl	OE.Configuration	OE.ECDAA	OE.Locality	OE.Credential	OE.Measurement	OE.FieldUpgradeInfo
T.Compromise				X				X						X			X											
T.Bypass																X	X											
T.Export					X											X							X					
T.Hack_Crypto		X																										
T.Hack_Physical				X																	X							
T.Imperson								X	X	X	X						X								X			
T.Import									X																			
T.Insecure_State						X	X									X							X					
T.Intercept					X				X											X								
T.Malfunction						X												X										
T.Modify				X				X		X							X											
T.Object_Attr_Change																X												
T.Replay																			X									
T.Repudiate_Transact													X															
T.Residual_Info														X														
T.Leak																					X							
Error! Reference source not found.	X																											
Error! Reference source not found.			X																				X					
Error! Reference source not found.				X												X												
Error! Reference source not found.											X													X				
OSP.RT_Measurement												X														X		
Error! Reference source not found.															X										X			
Error! Reference source not found.		X		X	X			X	X																			
Error! Reference source not found.																						X						X
A.Configuration																							X					

T.Compromise: An undetected compromise of the data in shielded locations may occur as a result of an attacker (whether an insider or outsider) attempting to perform actions that the individual or capability is not authorised to perform.

T.Compromise is countered by O.I&A, O.DAC, O.No_Residual_Info and O.Security_Roles. These objectives limit the ability of a user to the performance of only those actions that the user is authorised to perform:

- O.I&A: The TOE must identify all users, and shall authenticate the claimed identity except of the role "World" before granting a user access to the TOE facilities. This objective provides the prerequisite for the application of the access control roles for the subjects by uniquely identifying users and requiring authentication of the user bound to a subject.

- O.DAC: The TOE must control and restrict user access to the TOE protected capabilities and shielded locations in accordance with a specified access control policy where the object owner manages the access rights for their data objects using the principle of least privilege. This objective limits an attacker from performing unauthorised actions through a defined access control policy.

- O.No_Residual_Info: The TOE must ensure there is no "object reuse", i.e. there is no residual information in information containers or system resources upon their reallocation to different users.

- O.Security_Roles: The TOE must maintain security-relevant roles and association of users with those roles. This objective further supports the access control policy by associating each user with a role, which then can be assigned a specific access control policy.

T.Bypass: An unauthorised individual or user may tamper with TSF, security attributes or other data in order to bypass TOE security functions and gain unauthorised access to TOE assets.

T.Bypass is countered by O.Security_Attr_Mgt and O.Security_Roles. These objectives allow the TOE to invoke the TSF in all actions and to counter the ability of unauthorised users to tamper with TSF, security attributes or other data:

- O.Security_Attr_Mgt: The TOE must allow only authorised users to initialise and to change security attributes of objects and subjects. The management of security attributes shall support the principle of least privilege by means of role based administration and separation of duty. This objective requires that only authorised users be allowed to initialise and change security attributes, which counters the threat of an unauthorised user making such changes.

- O.Security_Roles: The TOE must maintain security-relevant roles and association of users with those roles.

T.Export: A user or an attacker may export data from shielded locations without security attributes or with insecure security attributes, causing the exported data to be erroneous and unusable, to allow erroneous data to be added or substituted for the original data, and/or to reveal secrets.

T.Export is countered by O.Export, O.Security_Attr_Mgt and OE.Configuration. These objectives ensure the protection of confidentiality and integrity of exported data with secure security attributes bound to these data.

- O.Export: When data are exported outside the TPM, the TOE shall securely protect the confidentiality and the integrity of the data as defined by the protected capability. The TOE shall ensure that the data security attributes being exported are unambiguously associated with the data.

- The objective O.Security_Attr_Mgt limits initialisation and management of security attributes of objects and subjects to authorised users only. The objective OE.Configuration requires the authorised user to manage these security attributes securely. Thus the object cannot be exported with insecure security attributes.

T.Hack_Crypto: Cryptographic key generation or operation may be incorrectly implemented, allowing an unauthorised individual or user to compromise keys generated within the TPM or encrypted data or undetected modification of data.

T.Hack_Crypto is countered by O.Crypto_Key_Man. The security objective ensures secure key management and cryptographic operation.

- O.Crypto_Key_Man: The TOE must manage cryptographic keys, including generation of cryptographic keys using the TOE random number generator as source of randomness, in a manner to protect their confidentiality and integrity.

T.Hack_Physical: An unauthorised individual or user of the TOE may cause unauthorised disclosure or modification of TOE assets by physically interacting with the TOE. The attacker may be a hostile user of the TOE.

T.Hack_Physical is countered by O.Tamper_Resistance and O.DAC: O.Tamper_Resistance requires the TOE to resist physical tampering of the TSF which control and restrict user access to the TOE protected capabilities and shielded locations according to O.DAC.

T.Imperson: An unauthorised individual may impersonate an authorised user of the TOE and thereby gain access to TOE data in shielded locations and protected capabilities.

T.Imperson is countered by O.I&A, O.Security_Roles, O.Import, O.Locality, OE.Locality, O.Limit_Actions_Auth These objectives prevent impersonation by authentication based on managed roles with their security attributes and access control considering security attributes of the users securely provided by the TOE environment:

- O.I&A: The TOE must identify all users, and shall authenticate the claimed identity except of the role “World” before granting a user access to the TOE facilities. This objective provides the prerequisite for the application of the access control roles for the subjects by uniquely identifying users and requiring authentication of the user bound to a subject.

- O.Security_Roles: The TOE must maintain security-relevant roles and association of users with those roles. This objective further supports the access control policy by associating each user with a role, which then can be assigned a specific access control policy.

- O.Import: When data are being imported into the TOE, the TOE must ensure that the data security attributes are being imported with the data and the data is from an authorised source. In addition, the TOE shall verify those security attributes according to the TSF access control rules. TOE supports the protection of confidentiality and the verification of the integrity of imported data.

- O.Locality includes locality as security attribute of the user to access control and OE.Locality ensures that trusted processes indicate their correct locality to the TPM and untrusted processes are able to assert the locality 0 or Legacy only to the TPM.

- O.Limit_Actions_Auth requires restricting the actions a user may perform before the TOE verifies the identity of the user.

T.Import: A user or attacker may import data without security attributes or with erroneous security attributes, causing key ownership and authorisation to be uncertain or erroneous and the system to malfunction or operate in an insecure manner.

T.Import is countered by O.Import, which states: When data are being imported into the TOE, the TOE must ensure that the data security attributes are being imported with the data and the data is from an authorised source. In addition, the TOE shall verify those security attributes according to the TSF access control rules. TOE supports the protection of confidentiality and the verification of the integrity of imported data. The integrity of the data in a sealed data blob is protected by the TOE.

T.Insecure_State: The TOE may start-up in an insecure state or enter an insecure state, allowing an attacker to obtain sensitive data or compromise the system.

T.Insecure_State is countered by O.Security_Attr_Mgt, O.Fail_Secure, O.General_Integ_Checks and OE.Configuration. These objectives ensure the integrity or secure security attributes and preservation of secure state in case of failure:

- O.Security_Attr_Mgt: The TOE must allow only authorised users to initialise and to change security attributes of objects and subjects. The management of security attributes shall support the principle of least privilege by means of role based administration and separation of duty.

- O.General_Integ_Checks: The TOE must provide checks on system integrity and user data integrity.

- O.Fail_Secure: The TOE must enter a secure failure mode in the event of a failure.

- OE.Configuration: This security objective requires the IT environment to install and configure the TOE for starting up in a secure way.

T.Intercept: An attacker may intercept the communication between a user and the TPM subjects to gain knowledge of the commands and data sent to the subject or manipulate the communication.

T.Intercept is directly countered by O.Sessions, which states: The TOE must provide the confidentiality of the parameters of the commands within an authorised session and the integrity of the audit log of the commands.

T.Intercept is countered by O.Import which states the TOE supports the protection of confidentiality and the verification of the integrity of imported data and by O.Export which states that when data are exported outside the TPM, the TOE must securely protect the confidentiality and the integrity of the data as defined for the protected capability.

T.Malfunction: TOE assets may be modified or disclosed to an unauthorised individual or user of the TOE, through malfunction of the TOE.

T.Malfunction is countered by O.Self_Test and O.Fail_Secure. These objectives address detection of and preservation of secure states in case of failure.

- O.Self_Test: The TOE must provide the ability to test itself, verify the integrity of the shielded data objects and the protected capabilities operate as designed and enter a secure state in case of detected errors.

- O.Fail_Secure: The TOE must enter a secure failure mode in the event of a failure.

T.Modify: An attacker may modify data in shielded locations or their security attributes in order to gain access to the TOE and its assets. The integrity of the information may be compromised due to the unauthorised modification or destruction of the information by an attacker.

T.Modify is countered by O.Limit_Actions_Auth, O.I&A, O.DAC and O.Security_Roles. These objectives support the ability of the TOE to limit unauthorised user access and to maintain data and system integrity through appropriate management of cryptographic data in particular:

- O.Limit_Actions_Auth: The TOE must restrict the actions a user may perform before the TOE verifies the identity of the user.

- O.I&A: The TOE must identify all users, and shall authenticate the claimed identity except of the role “World” before granting a user access to the TOE facilities.

- O.DAC: The TOE must control and restrict user access to the TOE protected capabilities and shielded locations in accordance with a specified access control policy where the object owner manages the access rights for their data objects using the principle of least privilege.

- O.Security_Roles: The TOE must maintain security-relevant roles and association of users with those roles.

T.Object_Attr_Change: A user or attacker may create an object with no security attributes or make unauthorised changes to security attribute values for an object to enable attacks.

T.Object_Attr_Change is directly countered by O.Security_Attr_Mgt, which states: The TOE shall allow only authorised users to initialise and to change security attributes of objects and subjects.

T.Replay: An unauthorised individual may gain access to the system and sensitive data through a “replay” or “man-in-the-middle” attack that allows the individual to capture identification and authentication data.

T.Replay is directly countered by O.Single_Auth, which states: The TOE must provide a single user authentication mechanism and require re-authentication to prevent “replay” and “man-in-the-middle” attacks.

T.Repudiate_Transact: An originator of data may deny originating the data to avoid accountability.

T.Repudiate_Transact is directly countered by O.MessageNR, which states: The TOE must provide user data integrity, source authentication, and the basis for source non-repudiation when exchanging data with a remote system.

T.Residual_Info: A user may obtain information that the user is not authorised to have when the data in shielded locations is no longer actively managed by the TOE (“data scavenging”).

T.Residual_Info is directly countered by O.No_Residual_Info, which states: The TOE must ensure there is no “object reuse”, i.e. there is no residual information in information containers or system resources upon their reallocation to different users.

T.Leak: An attacker may exploit information which is leaked from the TOE during usage of the TSF in order to disclose confidential assets.

T.Leak is countered by O.Tamper_Resistance: O.Tamper_Resistance requires the TOE to protect the assets against not only physical tampering but also leakage. Leakage may occur through but not limited to measures of electromagnetic emanations, variations in power consumption or by changes in processing time.

Error! Reference source not found.: **Error! Reference source not found.**

The **Error! Reference source not found.** is implemented by the O.Context_Management, which states: The TOE must ensure a secure wrapping of a resource (except seeds) in a manner that securely protects the confidentiality and the integrity of the data of this resource and allows the restoring of the resource on the same TPM and during the same operational cycle only - TPM operational cycle is a Startup Clear to a Shutdown Clear and contexts cannot be reloaded across a different Startup Clear to Shutdown Clear cycle from the one in which they are created.

OSP.ECDAA: The ECDAA issuer and the TPM owner establish a procedure for attestation without revealing the attestation information (i.e. the identity of the TPM).

The OSP.ECDAA is implemented by the security objectives O.ECDAA for the TOE and OE.ECDAA for the TOE environment. As a result, when a TPM authenticates to a verifier, the attestation information about the TPM is not revealed to the verifier.

- O.ECDAA: The TPM must support the TPM owner for attestation to the authenticity of measurement digests without revealing the attestation information by implementation of the TPM part of the ECDAA signing operation.

- OE.ECDAA: The DAA issuer must support a procedure for attestation without revealing the attestation information based on the ECDAA signing operation.

Error! Reference source not found.: The TPM supports multiple trusted processes obeying the principle of least privilege by means of role based administration and separation of duty.

The **Error! Reference source not found.** is implemented by the O.DAC and O.Security_Attr_Mgt. These objectives require the access control and the management of the security attributes to support delegation:

- O.DAC: The TOE must control and restrict user access to the TOE protected capabilities and shielded locations in accordance with a specified access control policy where the object owner manages the access rights for their data objects using the principle of least privilege.

- O.Security_Attr_Mgt: The TOE must allow only authorised users to initialise and to change security attributes of objects and subjects. The management of security attributes shall support the principle of least privilege by means of role based administration and separation of duty.

Error! Reference source not found.: The TCG platform supports multiple transitive trust chains by means of a mechanism known as locality. The Host Platform's trusted processes assert their locality to the TPM. The TPM shall guard access to resources PCRs and NV Storage Space, to keys and data to be imported, and to defined commands depending on the execution environment's privilege level.

The **Error! Reference source not found.** is implemented by the objective O.Locality and OE.Locality. These objectives address the TOE using locality for access control and the environment providing this security attribute of the user for the TOE.

- O.Locality: The TOE must control access to objects based on the locality of the process communicating with the TPM.

- OE.Locality: The developer of the host platform must ensure that trusted processes indicate their correct locality to the TPM and untrusted processes are only able to assert the locality 0 to the TPM.

Error! Reference source not found.

The root of trust for measurement calculates and stores the measurement digests as hash values of a representation of embedded data or program code (measured values) provided to the TPM by other parts of the root of trust for measurement.

The **Error! Reference source not found.** is implemented by the TOE and a platform part of the root of trust for measurement as follows.

- O.Record_Measurement: Describes the responsibility of the TOE: The TOE must support calculating hash values and recording the result of a measurement.

- OE.Measurement: Describes the responsibility of the platform part of the root of trust for measurement: The platform part of the root of trust for measurement provides a representation of embedded data or program code (measured values) to the TPM for measurement

Error! Reference source not found.: The root of trust for reporting reports on the contents of the RTS. A RTR reports is typically a digitally signed digest of the contents of selected values within a TPM (measurement, key properties or audit digest). The authenticity of the assets reported is based on the verification of the signature and the credential of the signing key.

The **Error! Reference source not found.** is implemented by the objectives

- O.Reporting: The TOE must report measurement digests and attest to the authenticity of measurement digests.

- OE.Credential: Addresses trustworthy procedures for creation of EK and AK credentials for root of trust for reporting.

Error! Reference source not found.: The TPM as root of trust for storage protects the assets (listed in Table 8 and Table 9) entrusted to the TPM in confidentiality and integrity.

The **Error! Reference source not found.** is implemented directly by the O.Crypto_Key_Man, O.Export and O.Import and supported by the O.I&A and O.DAC. These objectives require the protection of keys and data under Storage Root Key and the hierarchy of trust for storage outside the TOE:

- O.Crypto_Key_Man: The TOE must manage cryptographic keys, including generation of cryptographic keys using the TOE random number generator as source of randomness, in a manner to protect their confidentiality and integrity. This objective ensures the security of the key hierarchy used to protect the stored data.

- O.Export: When data are exported outside the TPM, the TOE must securely protect the confidentiality and the integrity of the data as defined for the protected capability. The TOE shall ensure that the data security attributes being exported are unambiguously associated with the data. This objective ensures the security of the data and their security attributes when exported to the storage outside the TOE.

- O.Import: When data are being imported into the TOE, the TOE must ensure that the data security attributes are being imported with the data and the data is from an authorised source. In addition, the TOE shall verify those security attributes according to the TSF access control rules. TOE supports the protection of confidentiality and the verification of the integrity of imported data. This objective ensures the security of the data and their security attributes when imported from storage outside the TOE.

- O.I&A: The TOE must identify all users, and shall authenticate the claimed identity except of the role “World” before granting a user access to the TOE facilities.. This objective ensures authentication and binding of user to the subjects performing export and import of the keys.

- O.DAC: The TOE must control and restrict user access to the TOE protected capabilities and shielded locations in accordance with a specified access control policy where the object owner manages the access rights for their data objects using the principle of least privilege. This objective addresses the access control for the objects.

Error! Reference source not found.: 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.

The **Error! Reference source not found.** is implemented by O.FieldUpgradeControl and OE.FieldUpgradeInfo:

- O.FieldUpgradeControl: Ensures that the field upgrade can only be performed by the Platform firmware and only authentic update data provided by the vendor are accepted.
- OE.FieldUpgradeInfo: The operational environment is required to ensure that the end user shall be aware of the field upgrade process and its result, whether the installed firmware is certified or not and the version of the certified TPM.

A.Configuration: The TOE will be properly installed and configured based on the instructions of the user guidance documentation (AGD).

The A.Configuration is directly covered by the objective for the TOE environment OE.Configuration, which states: The TOE must be installed and configured properly for starting up the TOE in a secure state. The security attributes of subjects and objects shall be managed securely by the authorised user.

6. Extended Components Definition

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

6.1 Family Random Number Generation

The family Random Number Generation (FCS_RNG) of the class FCS: cryptographic support describes the security functional requirements for random number generation used for cryptographic purposes. The random number generation is provided to the user and used internally, but it is not limited to generation of authentication data or cryptographic keys.

FCS_RNG Generation of random numbers

Family behavior

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

Component leveling:

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: FCS_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, 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

This section describes the security functional requirements (SFR) and the security assurance requirements (SAR) to be fulfilled by the TOE.

7.1 Security Functional Requirements

This section describes the SFR to be fulfilled by the TOE. It defines the subjects, objects and operations and introduces the notation for the operation of the SFR components.

7.1.1 Definitions of Subjects, Objects and TSF data

This section defines roles that subjects may use to access objects and their associated TSF data for authorisation. The roles USER, ADMIN and DUP are defined for objects and NV Index and operations that can be performed on or with that object or NV Index.

Table 7: Subjects

Subject	Description	TSF data
Platform firmware	Entity that controls the platform hierarchy	platformAuth, platformPolicy, security attributes: locality, physical presence if supported by the TOE[3]
Platform owner	Entity that controls the owner hierarchy	ownerAuth, ownerPolicy security attribute: locality
Privacy administrator	Entity that controls the endorsement hierarchy	endorsementAuth, endorsementPolicy security attribute: locality
Lockout administrator	Entity that controls the lockout mechanism of a TPM	lockoutAuth
USER	Entity that uses objects, keys, data in NV memory	authValue, authPolicy assigned to the object security attribute: locality
ADMIN	Entity that controls the certification of an object and changing the authValue of an object	authValue, authPolicy assigned to the object security attribute: locality
DUP	Entity that is allowed to	authValue for authorisation

[3] Support of physical presence is an optional feature of the TOE for authorization of the platform firmware.

Subject	Description	TSF data
	duplicate loaded objects	role DUP
World	Entity not authenticated	(none)

651 Table 8 defines Protected Objects that are user data or TSF data depending on the context
652 in which they are used, the operations applicable to these objects and their security
653 attributes.

654 **Table 8: Protected Objects, operations, security attributes and authorisation data**

#	Protected Objects	Operations	Security attributes
1	**Platform Hierarchy** Set of services to manage Platform firmware controls	**Seed** The PPS may be installed at manufacturing time or generated automatically on first boot **Disable** (cmd TPM2_HierarchyControl) **Change authorisation** (cmd TPM2_HierarchyChangeAuth) (cmd TPM2_SetPrimaryPolicy)	Authorisation data: **platformAuth**, hierarchy authorisation to change platform policy or authorisation and disable the platform hierarchy. **platformPolicy**, hierarchy policy authorisation to change the authorisation or policy for the Platform hierarchy Security attributes: **hierarchy proof**, secret value used to associate a hierarchy with tickets, objects or contexts **phEnable**, logical attribute which determines whether the hierarchy is enabled or disabled **phEnableNV,** logical attribute which determines whether platform hierarchy NV indices are enabled or disabled.
2	**Endorsement Hierarchy** Set of services to manage Privacy Administrator controls	**Seed** The EPS may be installed at manufacturing time or generated automatically on first boot **Enable/Disable** (cmd TPM2_HierarchyControl) **Change authorisation** (cmd TPM2_HierarchyChangeAuth) (cmd TPM2_SetPrimaryPolicy),	Authorisation data: **platformAuth**, hierarchy authorisation to enable/disable the Endorsement hierarchy. **platformPolicy**, hierarchy policy authorisation to enable/disable the Endorsement hierarchy. **endorsementAuth**, hierarchy authorisation to change the authorisation for the Endorsement hierarchy. **endorsementPolicy**, hierarchy policy authorisation to change the

#	Protected Objects	Operations	Security attributes
		(cmd TPM2_Clear)	authorisation for the Endorsement hierarchy Security attributes: **hierarchy proof**, secret value used to associate a hierarchy with tickets, objects or contexts **ehEnable**, logical attribute which determines whether the hierarchy is enabled or disabled
3	**Storage Primary Hierarchy** Set of services to manage Owner controls	**Seed** The SPS may be installed at manufacturing time or generated automatically on first boot **Clear** (cmd TPM2_Clear) **Enable/Disable** (cmd TPM2_HierarchyControl) **Change authorisation** (cmd TPM2_HierarchyChangeAuth, cmd TPM2_SetPrimaryPolicy)	Authorisation data: **platformAuth**, hierarchy authorisation to enable/disable the Storage Hierarchy or clear hierarchy objects. **platformPolicy**, hierarchy policy authorisation to enable/disable the Storage hierarchy or clear hierarchy objects. **ownerAuth**, hierarchy authorisation to use the Storage Primary Seed. **ownerPolicy**, hierarchy policy authorisation to use the Storage Primary Seed **lockoutAuth**, authorisation used to reset the dictionary attack protection Security attributes: **hierarchy proof**, secret value used to associate a hierarchy with tickets, objects or contexts **shEnable**, logical attribute which determines whether the hierarchy is enabled or disabled
4	**NULL hierachy** Set of services provided to user World	**Create** The nullSeed is set to a random value on every TPM reset.	nullProof
5	**Platform Primary**	**Create**	Authorisation data:

#	Protected Objects	Operations	Security attributes
	Object A root key created by the TPM that may be stored in the TPM or cached outside the TPM. The resource is instantiated in the Platform Hierarchy.	(cmd TPM2_CreatePrimary, cmd TPM2_CreateLoaded) **Delete** (cmd TPM2_EvictControl) **Make Persistent** (cmd TPM2_EvictControl)	**userAuth,** User auth secret value for the primary key **authPolicy**, digest representing a policy calculation **platformAuth**, hierarchy authorisation to use the Platform Primary Seed. **platformPolicy**, hierarchy policy authorisation to use the Platform Primary Seed Security attributes: **key template**, TPMT_PUBLIC, Part 2, §12.2.4, the public parameters used to create the key, set by the cmd TPM2_CreatePrimary or TPM2_CreateLoaded, may be restricted by the cmd TPM2_PolicyTemplate
6	**Endorsement Primary Key** A root key created by the TPM that may be stored in the TPM or cached outside the TPM. The resource is instantiated in the Endorsement Hierarchy.	**Create** (cmd TPM2_CreatePrimary, cmd TPM2_CreateLoaded) **Delete** (cmd TPM2_Clear) (cmd TPM2_EvictControl) **Make Persistent** (cmd TPM2_EvictControl)	Authorisation data: **userAuth,** User auth secret value **authPolicy**, digest representing a policy calculation **platformAuth**, hierarchy authorisation to use the Platform Primary Seed. **platformPolicy**, hierarchy policy authorisation to use the Platform Primary Seed **endorsementAuth**, hierarchy authorisation to use the Endorsement Primary Seed. **endorsementPolicy**, hierarchy policy authorisation to use the Endorsement Primary Seed **lockoutAuth**, authorisation used to reset the dictionary attack protection or clear Security attributes: **key template**, TPMT_PUBLIC, Part 2, §12.2.4, the public parameters

#	Protected Objects	Operations	Security attributes
			used to create the key, set by the cmd TPM2_CreatePrimary or TPM2_CreateLoaded, may be restricted by the cmd TPM2_PolicyTemplate
7	**Storage Primary Key** A root key created by the TPM that may be stored in the TPM or cached outside the TPM. The resource is instantiated in the Storage Hierarchy	**Create** (cmd TPM2_CreatePrimary, cmd TPM2_CreateLoaded) **Delete** (cmd TPM2_Clear) (cmd TPM2_EvictControl) **Make Persistent** (cmd TPM2_EvictControl)	Authorisation data: **userAuth,** User auth secret value **authPolicy**, digest representing a policy calculation **platformAuth**, hierarchy authorisation to use the Platform Primary Seed. **platformPolicy**, hierarchy policy authorisation to use the Platform Primary Seed **ownerAuth**, hierarchy authorisation to use the Storage Primary Seed. **ownerPolicy**, hierarchy policy authorisation to use the Storage Primary Seed **lockoutAuth**, authorisation used to reset the dictionary attack protection or clear Security attributes: **key template**, TPMT_PUBLIC, Part 2, §12.2.4, the public parameters used to create the key, set by the cmd TPM2_CreatePrimary or TPM2_CreateLoaded, may be restricted by the cmd TPM2_PolicyTemplate
8	**Context** Context are applicable to objects (User keys and Primary keys) and also sessions (authorisations and sequence).	**create** (cmd TPM2_ContextSave), **load** (cmd TPM2_ContextLoad) **delete** (cmd TPM2_FlushContext)	**hierarchy proof**, used as secret for authenticity and integrity **objectContextID** for transient and sequence objects **contextCounter** for sessions (for protection against replay attacks) **clearCount**: to avoid transient

#	Protected Objects	Operations	Security attributes
			object load after resume **resetValue**: to avoid context load after reset
9	**User Key** Any cryptographic key except the primary keys, i.e. ordinary or derived key.	**Create** (cmd TPM2_Create, cmd TPM2_CreateLoaded) **Make Persistent** (cmd TPM2_EvictControl) **Load** (cmd TPM2_Load, cmd TPM2_CreateLoaded) **Delete** (cmd TPM2_EvictControl)	Authorisation data: **userAuth,** User auth secret value **authPolicy**, digest representing a policy calculation **platformAuth**, hierarchy authorisation to use the Platform Primary Object. **platformPolicy**, hierarchy policy authorisation to use the Platform Primary Object **ownerAuth**, hierarchy authorisation to use the Storage Primary Key. **ownerPolicy**, hierarchy policy authorisation to use the Storage Primary Key Security attributes: **key template**, TPMT_PUBLIC, Part 2, §12.2.4, the public parameters used to create the key, set by the cmd TPM2_Create or TPM2_CreateLoaded
10	**PCR** Platform Configuration Register (PCR) intended to record measurement digests and to be used for attestation and access control.	**reset**: reset the PCR value to zero, if allowed for the specified PCR (cmd TPM2_PCR_ Reset), or set all PCR to their default initial condition or to their save state (cmd TPM2_Startup) **read**: read the value of all PCRs specified in pcrSelect (cmd TPM2_PCR_Read), **allocate**: set the desired PCR allocation of PCR and algorithms (cmd TPM2_PCR_Allocate),	Authorisation data: authValue, authPolicy Security attributes: All flags are defined in [8], sec. 6.14 TPM_PT_PCR TPM_PT_PCR_SAVE - indicates that the PCR is saved and restored by TPM_SU_STATE TPM_PT_PCR_EXTEND_L0, TPM_PT_PCR_EXTEND_L1, TPM_PT_PCR_EXTEND_L2, TPM_PT_PCR_EXTEND_L3, TPM_PT_PCR_EXTEND_L4

#	Protected Objects	Operations	Security attributes
		quote: hash the selected PCR, sign the value with an identified signing key and export it (cmd TPM2_Quote) **event**: calculate the hash value of the eventData and return the digests list, in case an implemented PCR is referenced an extend of the digests list is processed (cmd TPM2_PCR_Event), **extend**: calculate the hash value of the PCR value according the digests list or the result of a pending hash calculation (cmd TPM2_PCR_Extend and TPM2_EventSequenceComplete) and the interface commands _TPM_Hash_Start, _TPM_Hash_Data and _TPM_Hash_End defined in [8].	- indicates that the PCR may be extended from specific locality TPM_PT_PCR_RESET_L0, TPM_PT_PCR_RESET_L1, TPM_PT_PCR_RESET_L2, TPM_PT_PCR_RESET_L3, TPM_PT_PCR_RESET_L4 - indicates that the PCR may be reset by specific locality TPM_PT_PCR_NO_INCREMENT - indicates that modifications to this PCR will not increment the pcrUpdateCounter TPM_PT_PCR_DRTM_RESET - indicates that the PCR is reset by DRTM event
11	**NV storage** Non-volatile storage of the TPM provided to the user and protected by access rights managed by the TPM owner.	TPM2_NV_DefineSpace TPM2_NV_UndefineSpace TPM2_NV_UndefineSpaceSpecial TPM2_NV_Read TPM2_NV_ReadPublic TPM2_NV_Increment TPM2_NV_Extend TPM2_NV_SetBits TPM2_NV_Write TPM2_NV_ReadLock TPM2_NV_WriteLock TPM2_NV_ChangeAuthTPM2_NV_Certify TPM2_EvictControl	**TPM_NV_INDEX** Security attributes: platform controls (TPMA_NV_PPWRITE and TPMA_NV_PPREAD) owner controls (TPMA_NV_OWNERWRITE and TPMA_NV_OWNERREAD) user controls (TPMA_NV_AUTHREAD and TPMA_NV_AUTHWRITE) access policy (TPMA_NV_POLICYWRITE, authPolicy) additional security attributes: cf. [8], sec. 13.2, table 206, cf. [8] sec. 13.3, table 204
12	**RNG** The TPM random number generator	**read**: read the next random number generated by the TPM (cf. cmd	No security attributes

#	Protected Objects	Operations	Security attributes
	(RNG) creates random numbers provided to the user and for internal use (e.g. key generation, secrets, nonce).	TPM2_GetRandom), **refresh**: provides any data as input to the random number generator to refresh the internal state of the random number generator (cf. cmd TPM2_StirRandom)	
13	**Credentials** Data object containing encrypted credential information and the encryption key. It reflects the credential distribution for a key on a TPM. (cf. Credential Protection ch. 24).	**associate** of a credential with an object in a way that ensures that the TPM has validated the parameters of the credentialed object. (cmd TPM2_ActivateCredential) **create** the credential which was requested by the CA by encrypting the credential data and creating the credential encryption key (cmd TPM2_MakeCredential)	No security attributes
14	**Clock** Data object representing the TPM time value. It is a volatile value that increments each millisecond that the TPM is powered. A non-volatile value (NV Clock) is updated periodically from Clock.	**read:** get the current value of time (TPM2_ReadClock, TPM2_GetTime). **advance:** modify the value of the TPM's Clock (TPM2_ClockSet). **adjust:** modify the rate of advance of TPM's Clock (TPM2_ClockRateAdjust).	**resetCount:** non-volatile counter that is incremented on a successful TPM reset **restartCount:** non-volatile counter that is incremented when the TPM executes TPM Resume, TPM Restart or _TPM_Hash_Start **safe flag:** non-volatile flag to indicate that an orderly shutdown has occurred

655

656 7.1.2 Presentation of operations on SFR components

657 The CC allows several operations to be performed on functional requirements; *refinement*,
658 *selection*, *assignment*, and *iteration* are defined in chapter C.4 of Part 1 of the CC. Each of
659 these operations is used in this PP.

660 The **refinement** operation is used to add detail to a requirement, and thus further restricts
661 a requirement. Refinement of security requirements is denoted in the changed element in
662 **bold** text or is added to the component in a paragraph identified by the word "refinement"

and printed in bold text. In cases where words from a CC requirement were deleted, the corresponding words are crossed out ~~like this.~~

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the PP authors are denoted as underlined text and the original text of the component is given by a footnote. Selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made, [selection:], and are *italicised*.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the values of security attributes. Assignments that have been made by the PP authors are denoted by showing as underlined text and the original text of the component is given by a footnote. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:], and are *italicised*. If assignment is performed but require further selection or assignment the operation is printed as underlined text like this [selection:] or [assignment:], and the open operation is printed *italicised and underlined*.

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/" and the iteration indicator after the component identifier.

7.1.3 SFRs for the General Behavior of the TOE

This section contains SFRs that are relevant for the TOE in general or before it is in the operational state.

7.1.3.1 Management

FMT_SMR.1 Security roles

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

FMT_SMR.1.1 The TSF shall maintain the roles

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

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

Application note 1: The roles Platform firmware, Platform Owner and Privacy Administrator are defined for the hierarchies. The role Lockout Administrator is used to

[4] [assignment: *the authorised identified roles*]

reset lockout for authorisation value. The roles USER, ADMIN and DUP are defined for objects and NV Index.

FMT_SMF.1 Specification of Management Functions

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

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

(1) Management of hierarchies,
(2) Management of authorisation values,
(3) Management of security attributes of keys,
(4) Management of security attributes of PCR,
(5) Management of security attributes of NV storage areas,
(6) Management of security attributes of monotonic counters,
(7) Reset the Action Flag of TPM dictionary attack mitigation mechanism[5].

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 [assignment: *list of security attributes*].

FPT_STM.1 Reliable time stamps

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

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

Application note 2: The clock value of the TPM is not an actual universal time clock (UTC). The Clock is a volatile value that increments each millisecond that the TPM is fully powered. If the TPM is powered off or in sleep mode the Clock may not be running or the non-volatile value (NV Clock) may not be updated. It is the responsibility of the caller to associate the ticks to an actual UTC.

7.1.3.2 Data Protection and Privacy

FDP_RIP.1 Subset residual information protection

[5] [assignment: *list of management functions to be provided by the TSF*]

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[6] the following objects:
- SPS,
- Primary Keys,
- User keys,
- Context,
- PCR data,
- NV storage data where (TPMA_NV_PLATFORMCREATE == CLEAR)
- Credentials[7].

7.1.3.3 Cryptographic SFR

The TPM offers cryptographic primitives to be used on its external interfaces. Further, cryptographic algorithms are internally used in various situations. Although the TPM library specification defines identifiers for algorithms and parameter sets (where appropriate, see [8]), the concrete set of algorithms is not specified but platform and vendor specific. Hence, the corresponding SFRs (FCS_COP.1) contain open assignments that shall be performed by the ST writer dependent on the intended implementation.

The cryptographic key generation provides three different types of keys: ordinary keys, primary keys, and derived keys. Ordinary keys are generated from random bits: The output of the RNG is used to seed the computation of the secret keys that are stored in a shielded location of the TPM. Primary keys are generated from seed values that are usually persistently stored on the TPM. Derived keys are generated from the sensitive value of the parent key.

For the generation of keys, seeds and other sensitive data, two different schemes are specified ([7]), one for ECDH and one for all other uses. Both schemes use a hash based key derivation function (KDF), one is called KDFe, for ECDH, and the other KDFa. For the generation of primary keys, [7] specifies an additional scheme which uses a DRBG instantiated with a hierarchy seed. Based on the intended usage of the key, further processing may be required in order to get the appropriate form of the key.

FCS_RNG.1 Random number generation

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

FCS_RNG.1.1 The TSF shall provide a *[assignment: deterministic,* hybrid[8]] random number generator that implements: NIST SP 800-90A *[assignment: Hash_DRBG, HMAC_DRBG, CTR_DRBG] [18]*[9].

[6] [selection: *allocation of the resource to, deallocation of the resource from*]
[7] [assignment: *list of objects*]
[8] [selection: *physical, deterministic, hybrid*]
[9] [assignment: *list of security capabilities*]

FCS_RNG.1.2 The TSF shall provide random numbers that meet: Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG[10].

Application note 3: [7], section 11.4.10, describes the RNG in the TPM as hybrid random number generator (RNG), that produces seeds by an entropy source based on physical random processes and the seeds are used for a deterministic random bit generator complying to NIST SP 800-90A [18]. NIST SP 800-90A defines the three types of deterministic random bit generators listed in the SFR and ST author shall identify by assignment in the element FCS_RNG.1.1, which type is implemented in the TOE. The quality metric defined in the element FCS_RNG.1.2 will be fulfilled if the seeds have sufficient entropy and the assigned deterministic random number generator is correctly implemented. The Appendix 8.1 provides more details on evaluation of RNG. The RNG is used internally for generation of Primary Seeds, input to key generation, authorisation values and nonces.

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

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/PK The TSF shall generate cryptographic **primary [selection: *RSA, ECC, symmetric*]** keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [selection: *2048 bits, 256 bits, 384 bits, 128 bits*][11] that meet the following: TPM library specification [7], [8], [9], [assignment: *list of additional standards*].[12]

Application note 4: The two selections shall be performed consistently, i.e. if RSA is selected then the key size shall be 2048 bits, if ECC is selected then the key size shall be 256 bits and optionally 384 bits, if symmetric is selected then the key size shall be 128 bits and optionally 256 bits. If more than one primary key generation algorithm is supported by the TOE the ST writer shall iterate the component FCS_CKM.1/PK.

Application note 5: The ST author shall specify the used key generation algorithms and key sizes. The TPM library specification [7] defines two key derivation functions called KDFa and KDFe. They use a KDF in counter mode as specified in [22] with HMAC [16] as pseudorandom function. In addition, for the generation of primary keys, [7] defines a DRBG as specified in [18] used as pseudorandom function. In order to generate keys for dedicated algorithms, the generated values may need an appropriate post-processing. Examples for algorithm-specific post-processing are provided in the appendixes B and C of [7], other methods may also be used. The ST writer shall iterate the component FCS_CKM.1 if the TOE supports more than one key generation method.

Application note 6: The EPS and/or EK may be generated in the manufacturing environment and injected into the TOE. The manufacturer may only inject an EPS, however,

[10] [assignment: *a defined quality metric*]
[11] [assignment: *cryptographic key sizes*]
[12] [assignment: list of standards]

an EK cannot be injected without also injecting the EPS. This method is not addressed by this SFR.

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

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/RSA The TSF shall generate cryptographic **RSA** keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: TPM library specification [7], [8], [9], [assignment: *list of additional standards*].[13]

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

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/ECC The TSF shall generate cryptographic **ECC** keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: TPM library specification [7], [8], [9], [assignment: *list of additional standards*].[14]

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

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/SYMM The TSF shall generate cryptographic **symmetric** keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: TPM library specification [7], [8], [9], [assignment: *list of additional standards*].[15]

Application note 7: The refinements in the SFRs FCS_CKM.1/PK, FCS_CKM.1/ECC, FCS_CKM.1/RSA and FCS_CKM.1/SYMM are defined in order to specify the intended usage of the generated keys more precisely. The algorithms for the generation of these cryptographic keys are dependent on the intended usage of the keys.

FCS_CKM.4 Cryptographic key destruction

[13] [assignment: list of standards]
[14] [assignment: list of standards]
[15] [assignment: list of standards]

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 [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

Application note 8: FCS_CKM.4 destroys the cryptographic keys that were used by the operations as defined in FCS_COP.1. The ST author shall specify how the cryptographic keys are destroyed when not required anymore. A possible procedure may be the overwriting with fixed or random data.

FCS_COP.1/AES Cryptographic operation (symmetric encryption/decryption)

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

FCS_COP.1.1/AES The TSF shall perform symmetric encryption and decryption[16] in accordance with a specified cryptographic algorithm AES in the mode CFB [selection: *CTR, OFB, CBC, and ECB*][17] and cryptographic key sizes 128 [selection: *192, 256*] bits[18] that meet the following: NIST Pub 800-38a [23] or ISO/IEC 10116 [28] or ISO/IEC 18033-3 [32][19].

Application note 9: The TPM library specification [7], chapter 11.4.6, requires the TOE to implement AES in Cipher Feedback Mode (CFB) and allows support of the other block cipher modes listed for selection in the ST. The PC client specific interface specification [11] recommends that ECB mode should not be used. This selection may be empty. The selection of additional key sizes of AES may be empty.

[16] [assignment: *list of cryptographic operations*]
[17] [assignment: *cryptographic algorithm*]
[18] [assignment: *cryptographic key sizes*]
[19] [assignment: *list of standards*]

FCS_COP.1/SHA Cryptographic operation (hash function)

Hierarchical to: No other components.

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

FCS_COP.1.1/SHA The TSF shall perform hash value calculation[20] in accordance with a specified cryptographic algorithm SHA-1 and SHA-256 [selection: *SHA-384*][21] and cryptographic key sizes none[22] that meet the following: FIPS 180-4 [14][23].

Application note 10: The TPM shall implement an approved hash algorithm that has approximately the same security strength as its strongest asymmetric algorithm. If the TOE support additional hash functions the ST writer shall iterate the component FCS_COP.1 for these hash functions. The selection may be empty.

Application note 11: The usage of the hash algorithms by the TPM shall be implemented in accordance with NIST SP 800-107 [0].

FCS_COP.1/HMAC Cryptographic operation (HMAC calculation)

Hierarchical to: No other components.

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

FCS_COP.1.1/HMAC The TSF shall perform HMAC value generation and verification[24] in accordance with a specified cryptographic algorithm HMAC with SHA-1 and SHA-256 [selection: *SHA-384*][25] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: FIPS 198-1 [16] or ISO/IEC 9797-2 [27][26].

[20] [assignment: *list of cryptographic operations*]
[21] [assignment: *cryptographic algorithm*]
[22] [assignment: *cryptographic key sizes*]
[23] [assignment: *list of standards*]
[24] [assignment: *list of cryptographic operations*]
[25] [assignment: *cryptographic algorithm*]
[26] [assignment: *list of standards*]

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

Hierarchical to: No other components.

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

FCS_COP.1.1/RSAED The TSF shall perform asymmetric encryption and decryption[27] in accordance with a specified cryptographic algorithm RSA without padding, RSAES-PKCS1-v1_5, RSAES-OAEP[28] and cryptographic key sizes 2048 bit[29] that meet the following: PKCS#1v2.1 [26][30].

Application note 12: The TPM library specification part 2 [8] and 3 [9] define RSA encryption schemes

- RSA without padding: performing a modular operation with public key for encryption and private key for decryption on the message treated as unsigned integer without any padding (cf. command TPM2_RSA_Encrypt and TPM2_RSA_Decrypt with TPM_ALG_NULL in *keyhandle.scheme* and *inScheme* of the command).
- RSAES-PKCS1-v1_5 (cf. command TPM2_RSA_Encrypt and TPM2_RSA_Decrypt with TPM_ALG_RSAES in *keyhandle.scheme* and *inScheme* of the command)
    - for encryption: application of the padding algorithm RSAES-PKCS1-v1_5 to the message according to PKCS#1v2.1, chapter 7.2, and then performing a modular operation with public key of *keyHandle* on the padded message treated as unsigned integer.
    - for decryption: performing a modular operation with private key of *keyHandle* on the message treated as unsigned integer application, checking of the padding algorithm according to the message according to PKCS#1v2.1, chapter 7.2, and if padding is correct remove the padding for the decrypted message.
- RSAES-OAEP (cf. command TPM2_RSA_Encrypt and TPM2_RSA_Decrypt with TPM_ALG_OAEP in *keyhandle.scheme* and *inScheme* of the command)
    - for encryption: application of the padding algorithm RSAES-OAEP to the message according to PKCS#1v2.1, chapter 7.1, and then performing a modular operation with public key of *keyHandle* on the padded message treated as unsigned integer.
    - for decryption: performing a modular operation with private key of *keyHandle* on the message treated as unsigned integer application, checking of the padding algorithm according to the message according to PKCS#1v2.1,

[27] [assignment: *list of cryptographic operations*]
[28] [assignment: *cryptographic algorithm*]
[29] [assignment: *cryptographic key sizes*]
[30] [assignment: *list of standards*]

chapter 7.1, and if padding is correct remove the padding for the decrypted message.

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

Hierarchical to: No other components.

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

FCS_COP.1.1/RSASign The TSF shall perform signature generation and verification[31] in accordance with a specified cryptographic algorithm RSASSA_PKCS1v1_5, RSASSA_PSS[32] and cryptographic key sizes 2048 bit[33] that meet the following: PKCS#1v2.1 [26][34].

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

Hierarchical to: No other components.

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

FCS_COP.1.1/ECDSA The TSF shall perform signature generation and verification[35] in accordance with a specified cryptographic algorithm ECDSA with curve TPM_ECC_NIST_P256 [selection: *TPM_ECC_NIST_P384*], and [assignment: *other elliptic curve*][36] and cryptographic key sizes 256 [selection: *384*] bit[37] that meet the following: FIPS PUB 186-4 [15] or ISO/IEC 14888-3 [30][38].

Application note 13: The signature-creation is provided by the command TPM2_Sign and the signature verification is provided by the command TPM2_VerifySignature. The elliptic curve TPM_ECC_NIST_P256 is defined in FIPS PUB 186-4, section D.1.2.3. The optional curve TPM_ECC_NIST_P384 is defined in section D.1.2.4. The ST writer shall assign any other elliptic curve supported for signature creation and verification but this assignment may be empty if no other elliptic curve is supported.

FCS_COP.1/ECDAA Cryptographic operation (ECDAA commit)

---

[31] [assignment: *list of cryptographic operations*]
[32] [assignment: *cryptographic algorithm*]
[33] [assignment: *cryptographic key sizes*]
[34] [assignment: *list of standards*]
[35] [assignment: *list of cryptographic operations*]
[36] [assignment: *cryptographic algorithm*]
[37] [assignment: *cryptographic key sizes*]
[38] [assignment: *list of standards*]

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

FCS_COP.1.1/ECDAA The TSF shall perform signature generation[39] in accordance with a specified cryptographic algorithm ECDAA with curve *TPM_ECC_NIST_P256* and *TPM_ECC_BN_P256* [selection: *TPM_ECC_NIST_P384 and TPM_ECC_BN_P384*] [assignment: *other elliptic curve*][40] and cryptographic key sizes 256 [selection: *384*][41] that meet the following: TPM library specification [7][42].

Application note 14: The ECDAA sign operation is a modified Schnorr signature using ECDAA signing keys normally based on Barreto-Naehrig elliptic curve TPM_ECC_BN_P256 and optionally TPM_ECC_BN_P384_but the TOE may support other elliptic curves as well. The TPM_ECC_BN_P256 and TPM_ECC_BN_P384 are Barreto-Naehrig (BN) elliptic curves as defined in [ISO/IEC 15946-5: 2008 Clause 7.3 "BN curve"]. The first step of ECC anonymous signing operation is provided by command TPM2_Commit. The output is then used by command TPM2_Sign. The ST writer shall select TPM_ECC_NIST_P256 and TPM_ECC_BN_P256 and shall assign any other elliptic curve if supported for ECDAA. Both commands TPM2_Commit and TPM2_Sign shall use the same elliptic curve in order to run ECDAA protocol.

Application note 15: The ECDAA algorithm is not recognised by NIST as approved algorithm.

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

FCS_COP.1.1/ECDEC The TSF shall perform decryption of ECC key[43] in accordance with a specified cryptographic algorithm ECDH with curve [selection: *TPM_ECC_NIST_P256, TPM_ECC_NIST_P384, TPM_ECC_BN_P256, TPM_ECC_BN_P384],* [assignment: *other elliptic curve*][44] and cryptographic key sizes 256 bit [selection: *384 bit*][45] that that meet the following: TPM library

[39] [assignment: *list of cryptographic operations*]
[40] [assignment: *cryptographic algorithm*]
[41] [assignment: *cryptographic key sizes*]
[42] [assignment: *list of standards*]
[43] [assignment: *list of cryptographic operations*]
[44] [assignment: *cryptographic algorithm*]
[45] [assignment: *cryptographic key sizes*]

specification [7], NIST Special Publication 800-56A [20] or ISO/IEC 15946-1 [31][46].

Application note 16: The key decryption is implemented in the command TPM2_ECDH_ZGen.

7.1.3.4 Identification and Authentication SFR

The TPM identification and authentication capability is used to authorise the use of a Protected Object and Protected Capability. Note that the TCG Library Specification document refers to the identification and authentication process and access control as *authorisation.* Two basic mechanisms are provided for authentication:

- the prove of knowledge of a shared secret, i.e. password or a secret for HMAC, assigned to the entity as *authValue*; and
- the authentication of the user and verification of an intended state of the TPM and its environment encoded in *authPolicy* and assigned to the entity.

The authorisation may be for a command only or session based. The session type defines the used authorisation as HMAC session or policy session.

The *authValue* is linked to user roles. The *authValue* may be known or set to a randomly generated value. If the *authValue* is set to a randomly generated value it will be unknown to the user and the authentication is blocked. The *authPolicy* may be empty or set. If the *authPolicy* is set to 0 no authentication is possible. If the *authPolicy* is set it may require more than authentication of the user, cf. FIA_UAU.5.2 for the list of assertions a *authPolicy* may contain.

The session based authorisation uses *handles* and random *nonces*. The handle is assigned when the session is created and identifies the session until the session is closed. The session requires that a nonce shall be used only for one message and its reply. For instance, the TPM would create a nonce and send that in a reply. The requestor would receive that nonce (*nonceOlder*), generates its own nonce (*nonceNewer*) and includes both values in the calculation of the command-dependent authentication value. Then, the caller sends the command, the authentication value and *nonceNewer* to the TPM which checks the authentication value with the knowledge of both nonces and executes the command on success. The nonces link commands in the command chain and commands and responses.

Protected entities and their authentication data may be stored persistently in the TPM or outside the TPM. Note that cryptographic keys are considered as entities and do not undergo a special handling, hence this protection profile does not contain special requirements for the key management.

FIA_SOS.2 TSF Generation of secrets

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

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet uniform distribution of random variable generating the value.[47]

[46] [assignment: *list of standards*]

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

Application note 17: The TSF shall take the values to generate nonce from the RNG.

FMT_MSA.4/AUTH Security attribute value inheritance

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

FMT_MSA.4.1/AUTH The TSF shall use the following rules to set the value of security attributes:
(1) The bits userWithAuth and adminWithPolicy in the TPMA_OBJECT of an object are defined when the object is created and can never be changed.
(2) User authorised by policy session is allowed to change the authPolicy by means of command TPM2_PolicyAuthorize or TPM2_PolicyAuthorizeNV.[48]

Application note 18: The SFR FMT_MSA.4 describes management of authValue, which disables not only authentication data, but also management of authPolicy as security attributes for access control to objects.

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

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

FMT_MTD.1.1/AUTH The TSF shall restrict the ability to
(1) set[49] the platformAuth and platformPolicy[50] to the role Platform firmware[51];
(2) set[52] the endorsementAuth and endorsementPolicy[53] to the role Platform Owner[54];
(3) set[55] the endorsementAuth and endorsementPolicy [56] to the role Privacy Administrator[57],
(4) set by TPM2_Duplicate[58] the AuthValue or policyAuth of the object under the new parent to the same AuthValue or policyAuth of the duplicated object under the old parent[59] to the role DUP[60].

---

[47] [assignment: *a defined quality metric*]
[48] [assignment: *rules for setting the values of security attributes*]
[49] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[50] [assignment: *list of TSF data*]
[51] [assignment: *the authorised identified roles*]
[52] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[53] [assignment: *list of TSF data*]
[54] [assignment: *the authorised identified roles*]
[55] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[56] [assignment: *list of TSF data*]
[57] [assignment: *the authorised identified roles*]
[58] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[59] [assignment: *list of TSF data*]
[60] [assignment: *the authorised identified roles*]

(5) change[61] the lockout parameters (TPM2_DictionaryAttackParameters)[62] to the Lockout administrator[63].

FIA_AFL.1/Recover Authentication failure handling (recovery)

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication.

FIA_AFL.1.1/Recover The TSF shall detect when maxTries[64] of unsuccessful authentication attempts occur related to unsuccessful password or HMAC authentication attempts for

(1) objects where DA is active (i.e. noDA attribute is CLEAR)

(2) NV Index where DA is active (i.e. the TPMA_NV_NO_DA attribute is CLEAR)[65].

FIA_AFL.1.2/Recover When the defined number of unsuccessful authentication attempts has been met[66], the TSF shall block the authorisations for RecoveryTime seconds[67].

The counter failedTries is incremented when the authentication attempt failed.

The counter failedTries is decremented by one after recoveryTime seconds if:

(1) the TPM does not record an authorisation failure of a DA-protected entity,

(2) there is no power interruption, and

(3) failedTries is not zero.

The counter failedTries is reset to 0 by

(1) command TPM2_Clear()

(2) TPM2_DictionaryAttackLockReset() with lockoutAuth.

Application note 19: The refinement describes the failedTries behaviour the TPM can "self-heal" after a specified amount of time or be programmatically reset using proof of knowledge of an authorisation value.

FIA_AFL.1/Lockout Authentication failure handling (lockout)

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication.

FIA_AFL.1.1/Lockout The TSF shall detect when 1[68] unsuccessful authentication attempts occur related to failed authentication attempts with lockoutAuth using command TPM2_DictionaryAttackLockReset()[69].

---

[61] [selection: *change_default, query, modify, delete, [assignment: other operations]*]

[62] [assignment: *list of TSF data*]

[63] [assignment: *the authorised identified roles*]

[64] [assignment: *positive integer number*], *an administrator configurable positive integer within [assignment: range of acceptable values]*]

[65] [assignment: *list of authentication events*]

[66] [selection: *met, surpassed*]

[67] [assignment: *list of actions*]

[68][assignment: *positive integer number*], *an administrator configurable positive integer within [assignment: range of acceptable values]*]

[69] [assignment: *list of authentication events*]

FIA_AFL.1.2/Lockout When the defined number of unsuccessful authentication attempts has been met[70], the TSF shall block the TPM2_DictionaryAttackLockReset command for lockoutRecovery seconds[71].

FIA_AFL.1/PINPASS Authentication failure handling

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

FIA_AFL.1.1/PINPASS The TSF shall detect when pinCount[72] successful authentication events exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_PASS.
FIA_AFL.1.2/ PINPASS When the defined number of successful authentication events has been met[73], the TSF shall block further authorization attempts[74].

FIA_AFL.1/PINFAIL Authentication failure handling

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

FIA_AFL.1.1/PINFAIL The TSF shall detect when pinCount[75] unsuccessful authentication attempts exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_FAIL[76].
FIA_AFL.1.2/ PINFAIL When the defined number of unsuccessful authentication attempts has been met[77], the TSF shall block further authorization attempts[78].

FIA_UID.1 Timing of identification

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

FIA_UID.1.1 The TSF shall allow
(1) to execute indication _TPM_Hash_Start, _TPM_Hash_Data and _TPM_Hash_End,
(2) to execute commands that do not require authentication,
(3) to access objects where the entity owner has defined no authentication requirements (authValue, authPolicy),
(4) [assignment: other TSF-mediated actions][79]
on behalf of the user to be performed before the user is identified.

---

[70] [selection: *met, surpassed*]
[71] [assignment: *list of actions*]
[72] [assignment: *positive integer number*], *an administrator configurable positive integer within [assignment: range of acceptable values]*]
[73] [selection: *met surpassed*]
[74] [assignment: list of actions]
[75] [selection: *[assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]*]
[76] [assignment: *list of authentication events*]
[77] [selection: *met surpassed*]
[78] [assignment: list of actions]
[79] [assignment: *list of TSF-mediated actions*]

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

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1 The TSF shall allow

(1) to execute indication _TPM_Hash_Start, _TPM_Hash_Data and _TPM_Hash_End,

(2) to execute commands that do not require authentication,

(3) to access objects where the entity owner has defined no authentication requirements (authValue, authPolicy)[80]

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.

Application note 20: The commands that do not require authorisation are listed informatively in Table 11 of [7] and defined in [8].

FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5.1 The TSF shall provide

(1) Password based authentication mechanism,

(2) HMAC based authentication mechanism,

(3) Policy based authentication mechanism[81]

to support user authentication.

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

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

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

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

[80] [assignment: *list of TSF mediated actions*]

[81] [assignment: *list of multiple authentication mechanisms*]

(4) A HMAC or policy based authentication is required if the authHandle parameter of the command contain a valid handle of an authorisation session.
  (a) A HMAC based authentication is required if the authorisation session shall be created with a sessionType of TPM_SE_HMAC,
  (b) A policy based authentication is required if the authorisation session shall be created with a sessionType of TPM_SE_POLICY.

(5) A policy based authentication mechanism verifies that a policy session provides a sequence of policy assertions combined in logical AND and OR relations, which policyDigest matches the authPolicy associated with the object and the other conditions of a policy session context are fulfilled. The assertions may express conditions for
  (a) successful authentication with authValue defined for the authorised entity and the object to be accessed,
  (b) the command code of the authorised command to be executed,
  (c) the cpHash of the authorised command to be executed,
  (d) special condition for command TPM2_Duplicate(),
  (e) the locality of the authorised command to be executed,
  (f) the referenced object handle,
  (g) the current system time,
  (h) the content of the NV memory,
  (i) the value of selected PCR,
  (j) the assertion of physical presence if supported by the TOE,
  (k) the value of a shared secret,
  (l) the presence of a valid signature of the given parameters,
  (m) the value of the TPMA_NV_WRITTEN attribute of the specified NV index,
  (n) the value of the TPM_NT_PIN_PASS attribute of the specified NV index,
  (o) the value of the TPM_NT_PIN_FAIL attribute of the specified NV index,
  (p) the key template of the commands TPM2_CreatePrimary, TPM2_Create, and TPM2_CreateLoaded,
  (q) the validity of a Ticket.

  The TSF shall update the representation of the state of the TPM and its environment (policyDigest) on execution of the enhanced authorisation commands defined in [9] section 25. The result of the updated policyDigest shall depend on the called command and its dedicated parameters.

(6) The command TPM2_PolicyRestart shall reset a policy authorisation session to its initial state.[82]

Application note 21: The ST writer shall describe the implemented methods for physical presence authorisation if supported by the TOE. The Password based authentication mechanism can be used by human user because it does not need any cryptographic calculation for authentication as required in HMAC based authentication mechanism. The policy based authentication mechanism is described in [7], chapter 19. The *policyDigest* can be computed in a trial session simulating the policy session required

[82] [assignment: *rules describing how the multiple authentication mechanisms provide authentication*]

for authorisation, read from the TPM by means of the command TPM2_PolicyGetDigest and used as an object's *authPolicy*.

FIA_UAU.6 Re-authenticating

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

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions that multiple commands need to be executed in one authorisation session.[83]

FIA_USB.1 User-subject binding

Hierarchical to: No other components.
Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:
(1) the shared secret for the TPM objects to access (sessionKey),
(2) the handle of opened authentication session,
(3) the physical presence if supported by the TOE and asserted,
(4) the state of the TPM and its environment (policyDigest)[84].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:
(1) The TSF shall initialise the policyDigest value representing the state of the TPM and its environment with a zero digest (0…0). This shall take place at execution of the command TPM2_StartAuthSession[85].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:
(1) The TSF shall create the shared secret (sessionKey) and the session handle in case of a session based authorisation using the command TPM2_StartAuthSession.
(2) The TSF shall invalidate the shared secret (sessionKey) and the session handle in each of the following situations:
    (a) The command TPM2_FlushContext is executed for the corresponding session handle.
    (b) The flag continueSession of the session attributes is cleared.
    (c) The command TPM2_Startup is executed with the argument TPM_SU_CLEAR or TPM_SU_STATE.[86].

7.1.3.5 TSF Protection

FPT_TST.1 TSF testing

[83] [assignment: *list of conditions under which re-authentication is required*]
[84] [assignment: *list of user security attributes*]
[85] [assignment: *rules for the initial association of attributes*]
[86] [assignment: *rules for the changing of attributes*]

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

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

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

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

(3) [assignment: *further conditions under which self test should occur*][87]

to demonstrate the correct operation of sensitive parts of the TSF[88].

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of [assignment: *parts of TSF data*][89].

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of the TSF[90].

Application note 22: The ST writer shall define additional conditions in FPT_TST.1.1 in case that the TPM manufacturer implements additional self tests.

FPT_FLS.1/FS Failure with preservation of secure state (fail state)
Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_FLS.1.1/FS The TSF shall preserve a secure state **by entering the Fail state** when the following types of failures occur:
(1) If during TPM Restart or TPM Resume, the TPM fails to restore the state saved at the last Shutdown(STATE), the TPM shall enter Failure Mode and return TPM_RC_FAILURE.
(2) failure detected by TPM2_ContextLoad when the decrypted value of *sequence* is compared to the stored value created by TPM2_ContextSave(),
(3) failure detected by self-test according to FPT_TST.1,
(4) [assignment: *list of additional types of failures in the TSF*][91]

Application note 23: The ST writer shall perform the missing operation in the element FPT_FLS.1/FS according to the additional types of failures for which the TSF preserve a secure state if implemented by the TOE. The assignment may be "none" if no additional

---

[87] [selection: *during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions[assignment: conditions under which self test should occur]*]
[88] [selection: *[assignment: parts of TSF], the TSF*]
[89] [selection: *[assignment: parts of TSF data], TSF data*]
[90] [selection: *[assignment: parts of TSF], TSF*]
[91] [assignment: *list of types of failures in the TSF*]

types of failures are handled by the TSF. For case (2) in element FPT_FLS.1.1 refer to TPM spec part 3 chapter 11.3.1.

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

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1/SD The TSF shall preserve a secure state **by shutdown** when the following types of failures occur:

(1) detection of a physical attack,

(2) detection of environmental condition out of spec values[92].

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 [assignment: *additional physical tampering scenarios*][93] to the TSF[94] by responding automatically such that the SFRs are always enforced.

Application note 24: The ST writer shall perform the missing operation in the element FPT_PHP.3 by adding specific physical tampering scenarios for which resistance is claimed for the specific TOE. This assignment may be empty.

FDP_ITT.1 Basic internal transfer protection

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control.]

FDP_ITT.1.1 The TSF shall enforce the **TPM state control, TPM Object Hierarchy, Data import and export, Measurement and reporting, Access Control, NVM and Credential SFPs** [95]to prevent the disclosure[96] of user data when it is transmitted between physically-separated parts of the TOE

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

FPT_ITT.1 Basic internal TSF data transfer protection

[92] [assignment: *list of types of failures in the TSF*]
[93] [assignment: *physical tampering scenarios*]
[94] [assignment: *list of TSF devices/elements*]
[95] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[96] [selection : *disclosure, modification, loss of use*]

Hierarchical to: No other components.
Dependencies: No dependencies

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure[97] when it is transmitted between separate parts of the TOE.

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

7.1.4 SFRs Concerning the Object Hierarchy of the TOE

This section contains SFRs that affect the internal object hierarchy of the TOE.

7.1.4.1 TPM Operational States

The TOE internal states can be considered in different ways and abstraction levels. In this section the TPM is observed on the abstraction level as described in chapter 12 of [7]. Figure 3 summarises the states and state transitions of the TPM that are used in the subsequent SFRs. The introduced states can be explained as follows:

- Power-Off state: A hardware TPM is in power-off state when no power is applied to the TPM, or the power is on and a reset is being asserted. This state may be reached from any other state because power can be lost at any time. In that sense, Figure 3 is incomplete because not all possible state transitions are shown for clarity reasons. The TPM does not execute any function except transition to the Init state when it receives the _TPM_Init indication.
- Initialisation state: The TPM enters this state when it receives the _TPM_Init indication. This indication is provided in a platform-specific manner (cf. section 12.2.2 of [7] for details). In the Init state, only the commands TPM2_Startup and field upgrade and the indication of _TPM_Hash_Start, _TPM_Hash_Data and _TPM_Hash_End are accepted. All other commands do not change the state and imply an error return code. The TPM may perform self-test in the Init state and may enter Failure mode if the self-test detects any failure.
- FUM: The Field Upgrade Mode is described in the specification [7] in section 12.5 as an optional and vendor specific capability for upgrading the TPM firmware. The specification does not define the detailed behavior of Field Upgrade Mode and allows vendor specific implementation. According to the library specification the TPM enters the FUM from operational or Init state after receiving the command TPM2_FieldUpgradeStart and successful integrity and authenticity validation of the first upgrade data block, accepts TPM2_FieldUpgradeData commands only in FUM and exits FUM returning to normal operation or entering a mode that requires _TPM_Init before normal operations resume. The Field Upgrade Mode can also be reached after TPM_Init if Field upgrade loading process has been interrupted and needs to be resumed before the TPM returns to operational state. The TPM shall perform integrity and authenticity check, but may implement vendor specific

[97] [selection : *disclosure, modification*]

authorisation or vendor specific commands and related state transitions for FUM. The informative Figure 3 denotes these possible state transitions with dashed lines.

- Operational state: In this state the TPM was successfully initialised. The initialisation of the operational status of the TPM is done by the TPM2_Startup command and may restore a previously (by TPM2_Shutdown) saved status. Details are defined in section 12.2.3 and 12.2.4 of [7]. In that state, no restrictions of the accepted command set exists. Before the TPM may return a result based on a cryptographic algorithm, it is required to perform a specific self-test of that algorithm. If a command requires use of an untested algorithm or functional module, the TPM performs the test and then completes the command actions. This behavior is modeled in Figure 3 using a state transition to Self-Test.
  Please note that the TPM2_Shutdown command does not imply a reset nor any state change of the TPM: It is used to prepare the TPM for a power cycle and may be used to save the operational status of the TPM for a later restore. Details can be found in section 9.4 of [9].
- Self-Test state: This state implements the required tests of cryptographic algorithms and is not triggered by a dedicated TPM command. When performing a self-test on demand, the TPM should test only those algorithms needed to complete the command. The command TPM2_SelfTest may optionally cause the TPM to trigger a full self-test of all algorithms and functional blocks. Depending on the result, the TPM changes its state back to Operational or to Fail after completion of the self-test.
- Fail state: In Fail state the TPM does not allow any command except TPM2_GetTestResult and TPM2_GetCapability. The only way to exit Fail state is when it receives _TPM_Init.

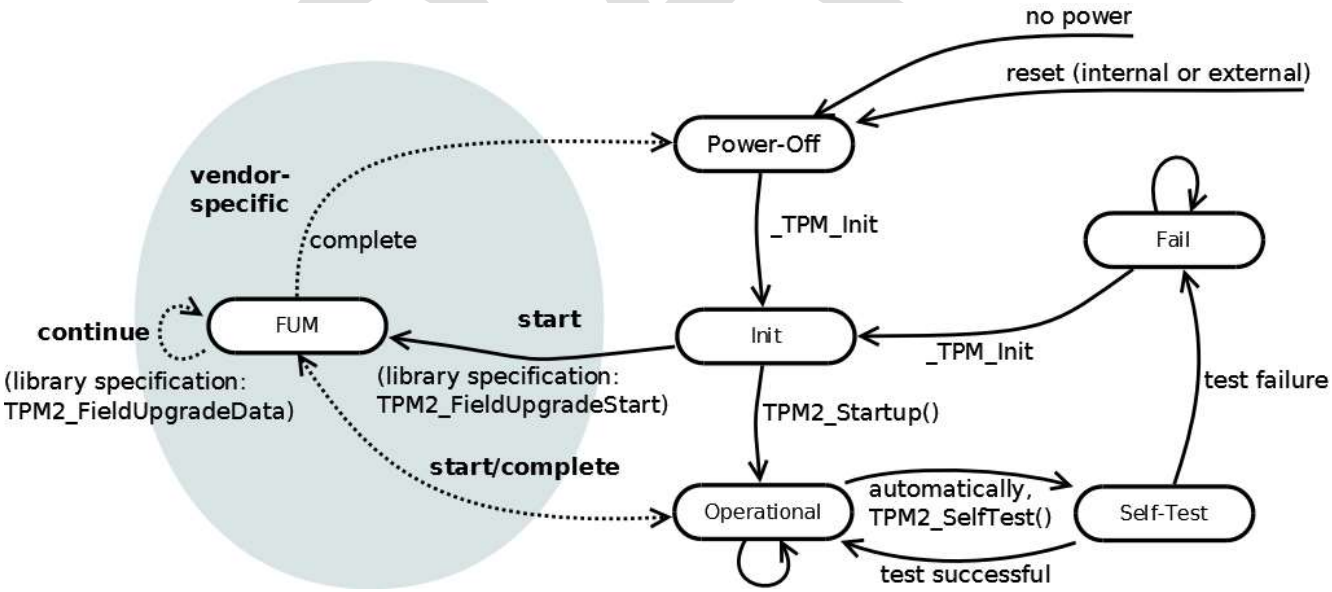

Figure 3: States of the TPM and its Transitions (informative)

Application note 25: Figure 3 illustrates the transitions between the TPM operational states as defined in the library specification, chapter 12 of [7]. The Field Upgrade Mode is vendor specific. The state transition and the commands TPM2_FieldUpgradeStart and TPM2_FieldUpgradeData shown in Figure 3 as described in the library specification are optional.

1398 The following table defines additional objects, operations and security attributes for the
1399 TPM state control SFP:

1400 **Table 9: Objects, operations and security attributes for the TPM state control SFP**

#	Protected Objects	Operations	Security attributes
1	**Shutdown BLOB** A set of variables that represent the operational status of the TPM as it is in the Operational state (see Figure 3).	**Generate** The shutdown BLOB is written to the NV memory by the command TPM2_Shutdown with parameter TPM_SU_STATE. **Resume** The shutdown BLOB is read from the NV memory by the command TPM2_Startup with parameter TPM_SU_STATE. The operational variables are restored with the values from the shutdown BLOB. This is called "TPM RESUME", see section 9.3 in [9]. **Restart** The shutdown BLOB is read from the the NV memory by the command TPM2_Startup with the parameter TPM_SU_CLEAR. Some operational variables are restored with the values from the shutdown BLOB. This is called "TPM RESTART", see section 9.3 in [9].	Security attributes: **Validation status**, used to check the validity of the Shutdown BLOB. After Generation of the Shutdown BLOB its validation status is positive. The execution of some commands may invalidate this status. The conditions that invalidate this validation status are defined in section 12.2.4 of [7]. In that document the BLOB is called "saved TPM state".
2	**Firmware update data** Data provided by the vendor in order to replace the firmware or parts of the firmware.	**TPM2_FieldUpgradeStart():** Entering FUM and accepting the first data block of Firmware update data **TPM2_FieldUpgradeData()** Read the following Firmware update data blocks.	Authorisation data for TPM2_FieldUpgradeStart(): **platformAuth, platformPolicy:** hierarchy authorisation to change platform policy or auth and disable the platform hierarchy. Security attributes of firmware update data: **Signature** over the first or the complete digest of Firmware update

#	Protected Objects	Operations	Security attributes
			data, generated by the TPM manufacturer **Digest** over each block or the complete Firmware update data

FDP_ACC.2/States Complete access control (operational states)

Hierarchical to: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1/States The TSF shall enforce the TPM State Control SFP[98] on all subjects and objects[99] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2/States The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1/States Security attribute based access control (operational states)

Hierarchical to: No other components.

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

FDP_ACF.1.1/States The TSF shall enforce the TPM State Control SFP[100] to objects based on the following

Subjects as defined in Table 7:

(1) Platform firmware with the security attributes platformAuth, platformPolicy and physical presence if supported by the TOE,

(2) all other subjects; their security attributes are irrelevant for this SFP,

Objects as defined in Table 8 and Table 9:

(1) Shutdown BLOB with the security attribute validation status,

(2) Firmware update data with security attributes signature of the TPM manufacturer and digest,

(3) all other objects; their security attributes are irrelevant for this SFP[101].

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

(1) The [assignment: *authorised role*] is authorised to change the TPM state to FUM if the authenticity of the first digest or the signature could be successfully verified.

(2) While in FUM state the Platform firmware is authorised to import or activate firmware data only after successful verification of its integrity and authenticity (see FDP_UIT.1/States).

---

[98] [assignment: *access control SFP*]

[99] [assignment: *list of subjects and objects*]

[100] [assignment: *access control SFP*]

[101] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

(3) The FUM state shall only be left when [assignment: *rules for a state transition from FUM to another state*].
(4) In the Init state the subject "World" is authorised to execute the commands TPM2_HashSequenceStart, TPM2_SequenceUpdate, TPM2_EventSequence-Complete, TPM2_SequenceComplete, TPM2_PCR_Extend, TPM2_Startup, TPM2_SelfTest, TPM2_GetRandom, TPM2_HierarchyControl, TPM2_Hierar-chyChangeAuth, TPM2_SetPrimaryPolicy, TPM2_GetCapability, TPM2_NV_Read, and the sequence _TPM_Hash_Start, _TPM_Hash_Data, and _TPM_Hash_End.
(5) In the Init state every subject is authorised to process the Resume operation on the Shutdown BLOB with state transition to Operational.
(6) In the Init state every subject is authorised to process the Restart operation on the Shutdown BLOB with state transition to Operational.
(7) In the Init state, if no Shutdown BLOB was generated or if the Shutdown BLOB is invalid (see attribute "Validation status") every subject is authorised to process the TPM2_Startup command. In case of the parameter TPM_SU_CLEAR the TPM shall change the state to Operational and initialise its internal operational variables to default initialisation values (Reset), otherwise the TPM shall return an error and stay in the same state.
(8) In the Operational state, nobody is authorised to execute the command TPM2_Startup. For all other subjects, objects and operations, the access control rules of the Access Control SFP shall apply (see FDP_ACF.1/AC).
(9) The Operational state shall change to Self-Test state if one of the commands TPM2_Selftest or TPM2_IncrementalSelfTest is executed or when a test of a dedicated functionality is required (see FPT_TST.1). In the Self-Test state, nobody is authorised to execute any other TPM command.
(10) The Self-Test state shall be left only after finishing the intended test of the dedicated functionality. In case of a successful test result the state shall change to Operational, otherwise to Fail.
(11) In the Fail state, every subject is authorised to execute the commands TPM2_GetTestResult and TPM2_GetCapability.
(12) In the Fail state the subject World is authorised to send a _TPM_Init indication with state change to Init.
(13) Any subject is authorised to prepare the TPM for a power cycle using the TPM2_Shutdown command and to create a shutdown BLOB by TPM2_Shutdown(TPM_SU_STATE).[102]

FDP_ACF.1.3/States The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].

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

[102] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

(1) Once the TPM receives a TPM2_SelfTest command and before completion of all tests, the TPM shall return TPM_RC_TESTING for any command that uses a command that requires a test.[103]

Application note 26: The ST writer shall define additional rules in FDP_ACF.1.2/States for the state transitions while the TPM is in FUM. Section 12.5 of [7] describes optional protected capabilities for upgrading the TPM firmware.

Application note 27: The _TPM_Init indication is normally signaled by the de-assertion of the TPM's reset signal. It may also be signaled by an interface protocol or setting.

Application note 28: When parts of the TSF or the complete TSF is replaced by a firmware update then the entire TOE needs to be considered as replaced by installation of another TOE.

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

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

FMT_MSA.1.1/States TSF shall enforce the TPM state control SFP[104] to restrict the ability to modify[105] the security attributes TPM state
(1) FUM[106] to Platform firmware[107],
(2) other than FUM[108] to any role[109].

Application note 29: The concrete restrictions in the TPM state control SFP to restrict the modification of the TPM state by dedicated roles is defined in FMT_MSA.1/States.

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

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

FMT_MSA.3.1/States The TSF shall enforce the TPM state control SFP[110] to provide restrictive[111] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/States The TSF shall allow ~~the~~ nobody[112] to specify alternative initial values to override the default values when an object or information is created.

---

[103] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].
[104] [assignment: *access control SFP(s), information flow control SFP(s)*]
[105] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[106] [assignment: *list of security attributes*]
[107] [assignment: *the authorised identified roles*]
[108] [assignment: *list of security attributes*]
[109] [assignment: *the authorised identified roles*]
[110] [assignment: *access control SFP, information flow control SFP*]
[111] [selection, choose one of*: restrictive, permissive, [assignment: other property]*]
[112] [assignment: *the authorised identified roles*]

FDP_UIT.1/States Data exchange integrity (operational states)

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]

FDP_UIT.1.1/States The TSF shall enforce the TPM state control SFP[113] to receive[114] ~~user~~ **firmware update** data in a manner protected from [selection: *modification, deletion, insertion, replay*] errors.

FDP_UIT.1.2/States The TSF shall be able to determine on receipt of ~~user~~ **firmware update** data, whether [selection: *modification, deletion, insertion, replay*] has occurred.

7.1.4.2 Creation and Modification of the TPM Hierarchy

Hierarchies are characterised by a parent-child relationship of objects. The TPM supports 3 hierarchies: the platform hierarchy, the storage hierarchy and the endorsement hierarchy. For (temporary) objects that are used only until the next TPM reset, a temporary object hierarchy may be created. The root of each TPM hierarchy is defined by a primary seed: Primary seeds are random values that are persistently stored in a TPM. The children of primary seeds are called primary objects.

Objects in a TPM hierarchy may be moved within the TPM hierarchy or even to a hierarchy of another TPM. This means that the moving object gets another parent object. In that case all children of the moving object including the whole sub-tree will move to the new position as well. The ability of objects to move is controlled by their attributes and can be restricted.

If the hierarchy is disabled the authValue and the authPolicy are not applicable.

FDP_SDI.1 Stored data integrity monitoring

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SDI.1.1 The TSF shall monitor user data stored in containers controlled by the TSF for data modifications and modification of hierarchy[115] on all objects, based on the following attributes: HMAC over the sensitive area of an object of the TPM hierarchy, object creation ticket[116].

Application note 30: The mentioned attributes in FDP_SDI.1 shall be generated at object creation time using the command TPM2_Create, TPM2_CreatePrimary, or TPM2_CreateLoaded. The HMAC over the sensitive data shall be done according to the section 27.6 of [7]. The object creation ticket (see section 10.6.3 of [8]) proves the environment in the object hierarchy at object creation time.

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

---

[113] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

[114] [selection: *transmit, receive*]

[115] [assignment: *integrity errors*]

[116] [assignment: *user data attributes*]

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

FDP_ACC.1.1/Hier The TSF shall enforce the TPM Object Hierarchy SFP[117] on

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

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

Operations
(1) TPM2_CreatePrimary,
(2) TPM2_CreateLoaded
(3) TPM2_HierarchyControl,
(4) TPM2_Clear,
(5) TPM2_ClearControl,
(6) TPM2_HierarchyChangeAuth,
(7) TPM2_SetPrimaryPolicy,
(8) TPM2_Load,
(9) TPM2_LoadExternal,
(10) TPM2_ReadPublic,
(11) Use.[118]

FDP_ACF.1/Hier Security attribute based access control (object hierarchy)
Hierarchical to: No other components.
Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/Hier The TSF shall enforce the TPM Object Hierarchy SFP[119] to objects based on the following:
Subjects:
(1) Platform firmware with security attribute authorisation state gained by authentication with platformAuth or platformPolicy,

[117] [assignment: *access control SFP*]
[118] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[119] [assignment: *access control SFP*]

(2) Platform owner with security attribute authorisation state gained by authentication with ownerAuth or ownerPolicy,
(3) Privacy administrator with security attribute authorisation state gained by authentication with endorsementAuth or endorsementPolicy,
(4) Lockout administrator with security attribute authorisation state,
(5) USER with authentication state gained with userAuth or authPolicy,
(6) World with no security attributes,
Objects:
(1) EPS,
(2) PPS,
(3) SPS,
(4) EK,
(5) PPO,
(6) SRK,
(7) Null Seed,
(8) object in a TPM hierarchy with security attributes: state of the hierarchy, fixedParent, fixedTpm[120]

FDP_ACF.1.2/Hier The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
(1) The subject World is authorised to create an EPS whenever the TPM is powered on and no EPS is present.
(2) The subject World is authorised to create an PPS whenever the TPM is powered on and no PPS is present.
(3) The subject World is authorised to create an SPS whenever the TPM is powered on and no SPS is present.
(4) The subject World is authorised to create a Null Seed whenever the TPM is reset.
(5) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the lockout administrator with lockoutAuth is authorised to change the SPS to a new value from the RNG (TPM2_Clear). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_Clear command.
(6) The Platform firmware is authorised to create a Platform Primary Object under PPS. The physical presence is not required if it is not if supported by the TOE or disabled for TPM2_CreatePrimary or TPM2_CreateLoaded command.
(7) The Platform owner is authorised to create a primary object (SRK) under SPS.
(8) The privacy administrator is authorised to create a primary object (EK) under EPS.
(9) The subject World is authorised to create temporary objects for no hierarchy (using the Null Seed).
(10) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the lockout administrator with lockoutAuth are authorised to remove all TPM context associated with a

[120] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

specific owner (TPM2_Clear). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_ClearControl command.

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

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

(13) The Platform firmware with platformAuth, platformPolicy or physical presence, if supported by the TOE the Platform owner and the privacy administrator are authorised to set the authorisation policy for the platform hierarchy (platformPolicy), the storage hierarchy (ownerPolicy) and the endorsement hierarchy (endorsementPolicy) using the command TPM2_SetPrimaryPolicy. The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_SetPrimaryPolicy command.[121]

FDP_ACF.1.3/Hier The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none[122].

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

(1) No subject is authorised to use any object of a hierarchy if the corresponding hierarchy is disabled (i.e phEnable for platform hierarchy is CLEAR, shEnable for Storage hierarchy is CLEAR, ehEnable for EPS hierarchy is CLEAR)[123].

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

[121] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[122] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[123] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

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

FMT_MSA.1.1/Hier TSF shall enforce the TPM Object Hierarchy SFP[124] to restrict the ability to modify[125] the security attributes fixedTPM and fixedParent[126] to nobody[127].

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

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

FMT_MSA.3.1/Hier The TSF shall enforce the TPM Object Hierarchy SFP[128] to provide restrictive[129] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Hier The TSF shall allow the creator of an object in a TPM hierarchy[130] to specify alternative initial values to override the default values when an object or information is created.

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

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

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

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

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

---

[124] [assignment: *access control SFP(s), information flow control SFP(s)*]
[125] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[126] [assignment: *list of security attributes*]
[127] [assignment: *the authorised identified roles*]
[128] [assignment: *access control SFP, information flow control SFP*]
[129] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[130] [assignment: *the authorised identified roles*]

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

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

(5) The Platform firmware with platformAuth, platformPolicy, or physical presence if supported by the TOE is authorised to enable the use of the Endorsement hierarchy and the Storage hierarchy (TPM2_HierarchyControl). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_HierarchyControl command.[131]

Application note 31: The TPM2_HierarchyControl command allows the security attributes *phEnable*, *shEnable*, and *ehEnable* to be changed when the proper authorisation is provided.

7.1.4.3 Data Import and Export

The TPM supports the creation of hierarchies of entities. A hierarchy is constructed with Storage Keys as the connectors to which other types of objects may be attached. Duplication is the process of allowing an object to be a child of additional parent keys. The new parent may be in a hierarchy of the same TPM or of a different TPM.

In order to summarise the correlations of different TPM commands regarding data import and export, Figure 4 illustrates possible scenarios: To be able to use an object as part of the TPM hierarchy, it needs to be previously loaded. The load operation is implemented as TPM2_Load, TPM2_CreateLoaded, or TPM2_LoadExternal command. The TPM2_Load command requires a TPM object that could have been created by TPM2_Create from an object template. TPM2_CreateLoaded combines creation and loading of an object in one command. The TPM2_LoadExternal command loads only the public area of an object (for example a public key) that could have been defined by an external system. If an object of the hierarchy of a TPM should be transferred into another TPM's object hierarchy, it needs to be duplicated based on the old object hierarchy first. Then it needs to be imported and later loaded based on the new object hierarchy, before it becomes part of the new hierarchy and can be used.

[131] [assignment: *rules for setting the values of security attributes*]

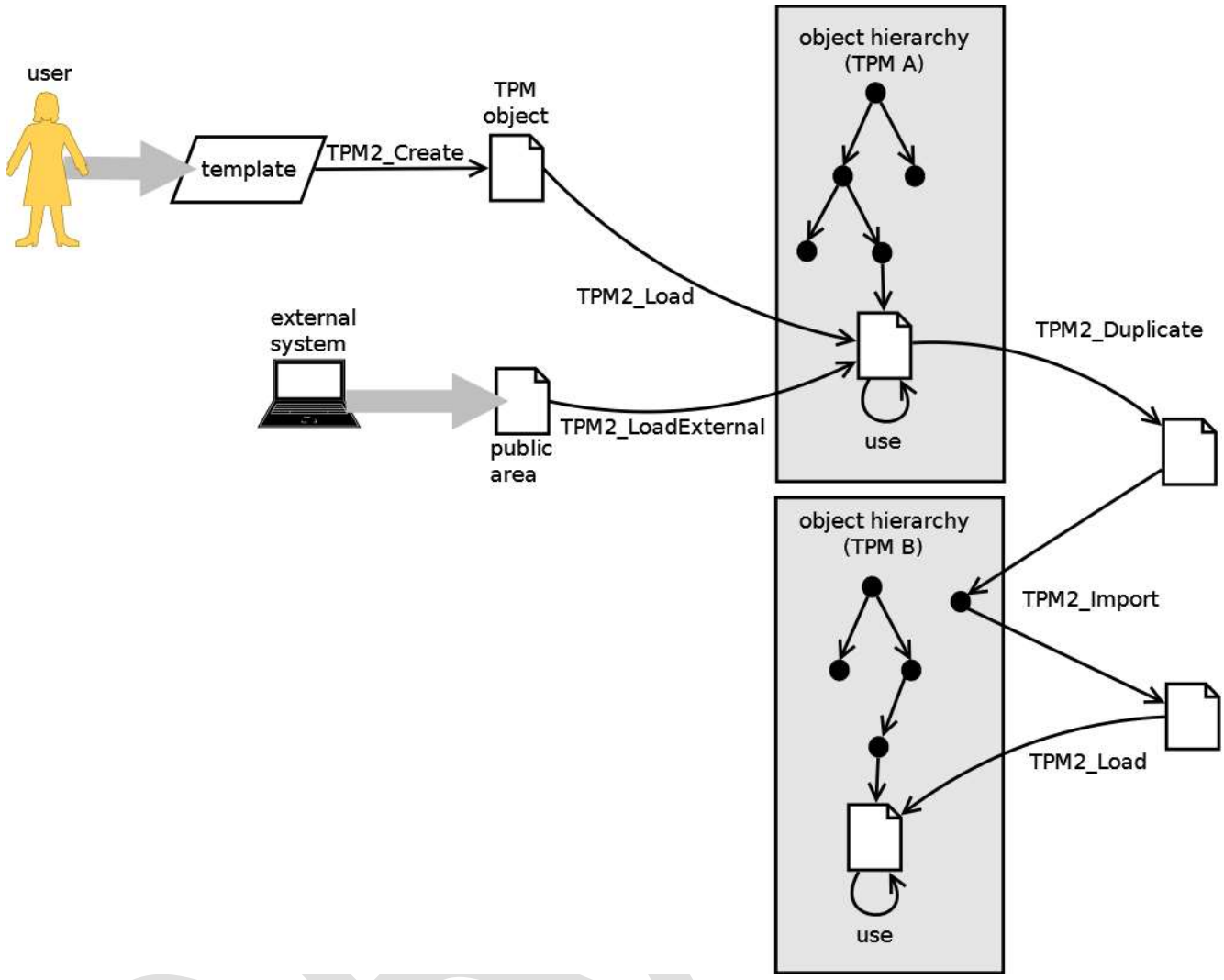

Figure 4: Object Export/Import Scenarios (informative)

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

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

FDP_ACC.1.1/ExIm The TSF shall enforce the Data Export and Import SFP[132] on
Subjects:
(1) USER,
(2) DUP,
(3) World
Objects:
(1) Platform Primary Object,
(2) Endorsement Primary Key,
(3) User Key,
(4) Context

[132] [assignment: *access control SFP*]

Operations
(1) duplicate by means of TPM2_Duplicate,
(2) export by means of TPM2_Create,
(3) load by means of TPM2_Load,
(4) export and load by means of TPM2_CreateLoaded
(5) load by means of TPM2_LoadExternal,
(6) import by means of TPM2_Import,
(7) unseal by means of TPM2_Unseal,
(8) save by means of TPM2_ContextSave
(9) load by means of TPM2_ContextLoad
(10) remove a context by means of TPM2_FlushContext[133]

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

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

FDP_ACF.1.1/ExIm The TSF shall enforce the Data Export and Import SFP[134] to objects based on the following:
Subjects:
(1) USER with authentication state gained with userAuth or authPolicy,
(2) DUP with authentication state gained with authPolicy,
(3) World without any successful authentication
Objects:
(1) Platform Primary Object with the security attributes platformAuth,
(2) Endorsement Primary Key with the security attributes authorisation data
(3) User Key with the security attributes authorisation data
(4) Context with the security attributes sequence number, hierarchy selector, HMAC[135]

FDP_ACF.1.2/ExIm The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
(1) The subject DUP is authorised to duplicate a loaded object under the following conditions:
  (a) the authorisation of the subject shall be provided in an authorisation session for duplication,
  (b) the object attribute "fixedParent" must not be set, and
  (c) the object attribute "nameAlg" must not be TPM_ALG_NULL.
(2) The subject USER is authorised to export an object using the TPM2_Create command.
(3) The subject USER authorised for the parent object is allowed to load objects into the TPM hierarchy using the command TPM2_Load.
(4) The subject USER is authorized to export and load an object using the TPM2_CreateLoaded command.

[133] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[134] [assignment: *access control SFP*]
[135] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

(5) The subject World authorised for the parent object is allowed to load objects into the TPM hierarchy using the command TPM2_LoadExternal.
(6) The subject USER authorised for the parent object is allowed to import an object using the TPM2_Import command under the following conditions:
  (a) The attributes "fixedTPM" and "fixedParent" of the object shall not be set.
  (b) If an encryption of the object to import is performed, then an integrity evidence value shall be part of the imported object.
  (c) If an integrity evidence value is present, the object shall only be imported after the integrity was successfully verified.
(7) The subject World is authorised to read the public portion of a TPM object using the command TPM2_ReadPublic.
(8) The subject USER is authorised to unseal a sealed data object using the TPM2_Unseal command.
(9) Every subject is authorised to save a context without authorisation.
(10) Every subject is authorised to load a saved context without authorisation if
  (a) the sequence number is in the accepted range,
  (b) the integrity of the context is successfully verified,
  (c) the TPM was not reset after the context saving and
  (d) the hierarchy associated with the context was not changed or disabled.
(11) Every subject is authorised to remove all context associated with a loaded object or session from the TPM memory (TPM2_FlushContext).[136]

FDP_ACF.1.3/ExIm The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none[137]

FDP_ACF.1.4/ExIm The TSF shall explicitly deny access of subjects to objects based on the following additional rules:
(1) No subject is authorised to move an object to another TPM's object hierarchy (using the duplicate and import operation) if the fixedTPM or the fixedParent attribute of that object is set.
(2) No subject is authorised to move an object to another position in a TPM object hierarchy (using the duplicate operation) if the fixedParent attribute of that object is set[138].

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

[136] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[137] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[138] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

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

FMT_MSA.1.1/ExIm TSF shall enforce the Data Export and Import SFP[139] to restrict the ability to use[140] the security attributes authorisation data[141] to every subject[142].

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

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

FMT_MSA.3.1/ExIm The TSF shall enforce the Data Export and Import SFP[143] to provide restrictive[144] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/ExIm The TSF shall allow ~~the~~ nobody[145] to specify alternative initial values to override the default values when an object or information is created.

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

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

FDP_ETC.2.1/ExIm The TSF shall enforce the Data Export and Import SFP[146] when exporting user data, controlled under the SFP(s), outside of the TOE.

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

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

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

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

---

[139] [assignment: *access control SFP(s), information flow control SFP(s)*]
[140] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[141] [assignment: *list of security attributes*]
[142] [assignment: *the authorised identified roles*]
[143] [assignment: *access control SFP, information flow control SFP*]
[144] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[145] [assignment: *the authorised identified roles*]
[146] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

(2) The sensitive area of an object from the TPM hierarchy shall be symmetrically encrypted before its export using the command TPM2_Create or TPM2_CreateLoaded. The used key and the IV should be derived from the secret seed of the parent in the TPM hierarchy.
(3) An exported context (using the command TPM2_ContextSave) shall be symmetrically encrypted and integrity protected with a HMAC.
(4) When exporting an object using the command TPM2_Duplicate then the following actions shall be performed:
  (a) If the encryptedDuplication attribute is set or the caller provides a symmetric algorithm then the sensitive part of the data shall be symmetrically encrypted and integrity protected (called: inner duplication wrapper).
  (b) If the encryptedDuplication attribute is set or the caller provides a new parent in a TPM hierarchy then the inner duplication wrapper shall be symmetrically encrypted and integrity protected (called outer duplication wrapper). The used key shall be derived from a seed that shall be asymmetrically encrypted with the public key of the intended new parent in the TPM object hierarchy.[147]

Application note 32: The details of the derivation of the key and IV for the symmetric encryption and HMAC generation for export of the sensitive area of objects are specified in section 22.4 and 22.5 of [7].

Application note 33: The details of the derivation of the key and IV for the symmetric encryption and HMAC generation for export of contexts are specified in section 30.3 of [7].

Application note 34: The details of the inner duplication wrapper for the TPM2_Duplicate command are defined in section 23.3.2 of [7].

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

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

FDP_ITC.2.1/ExIm The TSF shall enforce the Data Export and Import SFP[148] when importing user data, controlled under the SFP, from outside of the TOE.

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

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

[147] [assignment: *additional exportation control rules*]
[148] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

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

FDP_ITC.2.5/ExIm The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:
(1) If an inner or an outer wrapper is present then a valid integrity value shall be present.[149]

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

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

FDP_UCT.1.1/ExIm The TSF shall enforce the Data Export and Import SFP[150] to transmit[151] user data in a manner protected from unauthorised disclosure.

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

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

FDP_UIT.1.1/ExIm The TSF shall enforce the Data Export and Import SFP[152] to transmit and receive[153] user data in a manner protected from modification[154] errors.

FDP_UIT.1.2/ExIm The TSF shall be able to determine on receipt of user data, whether modification[155] has occurred.

7.1.4.4 Measurement and Reporting

An integrity measurement is a value that represents a possible change in the trust state of the platform. The TPM supports this measurement using the extension of an accumulative hash in a PCR. Integrity reporting is the process of attesting integrity measurements recorded in a PCR. PCR may also be used to gate access to an object. If selected PCR do not have the required values, the TPM will not allow use of the object. A TPM may maintain multiple banks of PCR. A PCR bank is a collection of PCR that are extended with the same hash algorithm.

Another aspect of measurement and reporting is the concept of tickets: A ticket is a HMAC signature that uses a proof value as the HMAC key. It is used as a replacement of an

[149] [assignment: *additional importation control rules*]
[150] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[151] [selection: *transmit, receive*]
[152] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[153] [selection: *transmit, receive*]
[154] [selection: *modification, deletion, insertion, replay*]
[155] [selection: *modification, deletion, insertion, replay*]

asymmetric digital signature in order to avoid the required computational effort of asymmetric operations.

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

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/M&R The TSF shall enforce the Measurement and Reporting SFP[156] on

subjects

(1) Platform firmware,

(2) USER,

(3) ADMIN,

(4) World,

objects

(1) PCR,

(2) TPM objects,

operations

(1) TPM2_PCR_Allocate,

(2) TPM2_PCR_Reset,

(3) TPM2_PCR_Extend,

(4) TPM2_PCR_Event,

(5) TPM2_PCR_Read,

(6) TPM2_Quote,

(7) TPM2_CertifyCreation[157]

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

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/M&R The TSF shall enforce the Measurement and Reporting SFP[158] to objects based on the following:

Subjects:

(1) Platform firmware with security attribute authorisation state gained by authentication with platformAuth or platformPolicy or locality,

(2) USER with authentication state gained with authValue or authPolicy,

(3) ADMIN with authentication state gained with authValue or authPolicy,

(4) World with no security attributes,

Objects:

(1) PCR with the security attribute PCR-attributes TPM_PT_PCR,

(2) TPM objects with the security attributes authentication data (authValue, authPolicy)[159]

---

[156] [assignment: *access control SFP*]

[157] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[158] [assignment: *access control SFP*]

[159] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

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

(1) The Platform firmware platformAuth, platformPolicy or with physical presence if supported by the TOE is authorised to set the desired PCR allocation of the PCR and the algorithms (TPM2_PCR_Allocate). The physical presence is not required if it is not supported by the TOE or disabled for TPM2_PCR_Allocate command.

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

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

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

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

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

(7) Authorised subjects of role USER are allowed to prove the association between an object and its creation data by creation of a ticket using the command TPM2_CertifyCreation. The authorisation shall be done based on the key that is used to sign the attestation block.[160]

FDP_ACF.1.3/M&R The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none[161].

FDP_ACF.1.4/M&R The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[162].

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

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/M&R TSF shall enforce the Measurement and Reporting SFP[163] to restrict the ability to modify[164] the security attributes PCR attributes, PCR extension algorithm, used hash algorithm[165] to Platform firmware[166].

---

[160] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[161] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[162] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

[163] [assignment: *access control SFP(s), information flow control SFP(s)*]

[164] [selection: *change_default, query, modify, delete, [assignment: other operations]*]

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

Hierarchical to: No other components.

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

FMT_MSA.3.1/M&R The TSF shall enforce the Measurement and Reporting SFP[167] to provide restrictive[168] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/M&R The TSF shall allow ~~the~~ nobody[169] to specify alternative initial values to override the default values when an object or information is created.

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

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FCO_NRO.1.1/M&R The TSF shall be able to generate evidence of origin for transmitted attestation structure (TPM2B_ATTEST) and object creation tickets[170] at the request of the originator[171].

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

(1) magic number for identification whether the TPM produced the signed digest or any external entity,
(2) type of the attestation structure indicating the contents of the attested parameter,
(3) qualified name of the key used to sign the attestation data (qualifiedSigner),
(4) external information supplied by the caller,
(5) values of clock, resetCount, restartCount and Safe,
(6) the firmware version[172]

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

(1) PCR data (using the command TPM2_Quote), or
(2) audit digests (using the command TPM2_GetSessionAuditDigest), or
(3) a ticket that was produces by the TPM (using the command TPM2_CertifyCreation)[173]

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[174] given as soon as the recipient can verify the signature and has confidence to the key that is used to sign[175].

---

[165] [assignment: *list of security attributes*]
[166] [assignment: *the authorised identified roles*]
[167] [assignment: *access control SFP, information flow control SFP*]
[168] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[169] [assignment: *the authorised identified roles*]
[170] [assignment: *list of information types*]
[171] [selection: *originator, recipient, [assignment: list of third parties]*]
[172] [assignment: *list of attributes*]
[173] [assignment: *list of information fields*]

Application note 35: The key used for signing may be any key with the sign attribute set. If a key is not restricted to a dedicated scheme then the caller of the corresponding command may indicate the signing scheme to be used. If an anonymous scheme (TPM_ALG_ECDAA) is used for signing, the qualifiedSigner parameter of the corresponding command shall be an empty buffer.

Application note 36: If the used signature key is not in the endorsement or platform hierarchy, then the mentioned attribute values resetCount, restartCount and firmwareVersion shall be obfuscated according to section 20 of [9] for privacy protection reasons.

7.1.5 SFRs for the TOE Operation

7.1.5.1 Access SFR

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

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/AC The TSF shall enforce the Access Control SFP[176] on

subjects

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

objects

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

operations

(1) TPM2_EvictControl,
(2) TPM2_ClockSet,
(3) TPM2_ClockRateAdjust,
(4) TPM2_ReadClock,
(5) TPM2_GetTime,
(6) TPM2_VerifySignature,
(7) TPM2_Sign,
(8) TPM2_GetRandom,
(9) TPM2_StirRandom,
(10) TPM2_RSA_Encrypt,
(11) TPM2_RSA_Decrypt,

---

[174] [selection: *originator, recipient, [assignment: list of third parties]*]
[175] [assignment: *limitations on the evidence of origin*]
[176] [assignment: *access control SFP*]

(12) TPM2_ECDH_KeyGen,
(13) TPM2_ECDH_ZGen,
(14) TPM2_ECC_Parameters,
(15) TPM2_HMAC_Start,
(16) TPM2_HashSequenceStart,
(17) TPM2_SequenceUpdate,
(18) TPM2_SequenceComplete,
(19) TPM2_EventSequenceComplete,
(20) TPM2_HMAC,
(21) TPM2_Hash[177]

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

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

FDP_ACF.1.1/AC The TSF shall enforce the Access Control SFP[178] to objects based on the following

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

Objects:
(1) User key with security attributes TPM_ALG_ID, TPMA_OBJECT,
(2) TPM objects,
(3) Clock with security attributes: resetCount, restartCount, safe-flag,
(4) Data with security attribute "externally provided"[179].

FDP_ACF.1.2/AC The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
(1) The Platform firmware platformAuth, platformPolicy or with physical presence if supported by the TOE and the Platform owner are authorised to control the persistence of loadable objects in TPM memory

[177] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[178] [assignment: *access control SFP*]

[179] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

(TPM2_EvictControl). The physical presence is not required if it is not supported by the TOE or disabled for TPM2_EvictControl command.

(2) The Platform firmware platformAuth, platformPolicy or with physical presence if supported by the TOE and the Platform owner are authorised to advance the value and to adjust the rate of advance of the TPMs clock (TPM2_ClockSet, TPM2_ClockRateAdjust). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_ClockSet respective TPM2_ClockRateAdjust command.

(3) Any subject is authorised to get the current value of time, clock, resetCount and restartCount (TPM2_ReadClock).

(4) A subject with the role USER endorsed by the Privacy administrator or the keyHandle identifier of a loaded key that can perform digital signatures is authorised to get the current value of time and clock (TPM2_GetTime).

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

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

(7) A subject with the role USER is authorised to generate digital signatures using the command TPM2_Sign for externally provided data (hash). The user authorisation shall be done based on the required authorisation of the key that will perform signing. The key attributes shall allow the signing operation for externally provided data.

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

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

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

(11) Any subject is authorised to perform RSA encryption using the command TPM2_RSA_Encrypt for externally provided data. The key attributes shall allow the encrypt operation for externally provided data.

(12) A subject with the role USER is authorised to perform RSA decryption using the command TPM2_RSA_Decrypt for externally provided data. The user authorisation shall be done based on the required authorisation of the key that will be used for decryption. The key attributes shall allow the decrypt operation for externally provided data.

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

(14) A subject with the role USER is authorised to recover a value that is used in ECC based key sharing protocols using the command TPM2_ECDH_ZGen. The user authorisation shall be done based on the required authorisation of the involved private key.

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

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

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

(18) The subject USER is authorised to add data to a hash, event or HMAC sequence using the command TPM2_SequenceUpdate.
(19) The subject USER is authorised to add the last part of data (if any) to a hash or HMAC sequence using the command TPM2_ SequenceComplete.
(20) The subject USER is authorised to add the last part of data (if any) to an event sequence using the command TPM2_EventSequenceComplete.
(21) Any subject is authorised to perform hash operations on a data buffer using the command TPM2_Hash.
(22) A subject with the role USER is authorised to perform HMAC operations on a data buffer. The user authorisation shall be done based on the required authorisation of the involved symmetric key.
(23) A subject with the role USER is authorised to generate HMACs using the command TPM2_HMAC for externally provided data (hash). The user authorisation shall be done based on the required authorisation of the key that will perform the HMAC. The key attributes shall allow the signing operation for externally provided data. [180]

FDP_ACF.1.3/AC The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*][181]

FDP_ACF.1.4/AC The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

FMT_MSA.1/AC Management of security attributes (access control)

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

FMT_MSA.1.1/AC TSF shall enforce the Access Control SFP[182] to restrict the ability to

(1) query[183] the security attributes digital signature of the audit session digest (TPM2_GetSessionAuditDigest)[184] to privacy administrator[185]

(2) query[186] the security attributes TPMT_PUBLIC_PARMS[187] (TPM2_TestParms) to World[188].

[180] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[181] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[182] [assignment: *access control SFP(s), information flow control SFP(s)*]
[183] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[184] [assignment: *list of security attributes*]
[185] [assignment: *the authorised identified roles*]
[186] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[187] [assignment: *list of security attributes*]

(3) query[189] the security attributes TPMS_ALGORITHM_DETAILS_ECC[190] (TPM2_ECC_Parameters) to World[191].

(4) increment[192] the security attributes resetCount and restartCount[193] to every subject[194],

(5) reset[195] the security attributes resetCount, restartCount and the safe-flag of the TPM Clock[196] by means of command TPM2_Clear to Platform firmware authorised by platformAuth, platformPolicy or physical presence (if supported by the TOE) and the lockout administrator[197],

(6) if supported by the TOE: change[198] the security attribute Physical Presence requirement for all commands in the setList of TPM2_PP_Comands to "required" and all commands in the clearList to "not required" of TPM2_PP_Comands[199] to Platform firmware authorised by platformAuth, platformPolicy or physical presence[200],

(7) change[201] the security attributes authorisation secret (authValue) of TPM objects (TPM2_ObjectChangeAuth)[202] to ADMIN[203].

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

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

FMT_MSA.3.1/AC The TSF shall enforce the Access Control SFP[204] to provide restrictive[205] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/AC The TSF shall allow the USER, ADMIN[206] to specify alternative initial values to override the default values when an object or information is created.

---

188 [assignment: *the authorised identified roles*]
189 [selection: *change_default, query, modify, delete, [assignment: other operations]*]
190 [assignment: *list of security attributes*]
191 [assignment: *the authorised identified roles*]
192 [selection: *change_default, query, modify, delete, [assignment: other operations]*]
193 [assignment: *list of security attributes*]
194 [assignment: *the authorised identified roles*]
195 [selection: *change_default, query, modify, delete, [assignment: other operations]*]
196 [assignment: *list of security attributes*]
197 [assignment: *the authorised identified roles*]
198 [selection: *change_default, query, modify, delete, [assignment: other operations]*]
199 [assignment: *list of security attributes*]
200 [assignment: *the authorised identified roles*]
201 [selection: *change_default, query, modify, delete, [assignment: other operations]*]
202 [assignment: *list of security attributes*]
203 [assignment: *the authorised identified roles*]
204 [assignment: *access control SFP, information flow control SFP*]
205 [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
206 [assignment: *the authorised identified roles*]

Application note 37: The default values are defined on object creation using the command TPM2_Create, TPM2_CreatePrimary, or TPM2_CreateLoaded.

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

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

FDP_UCT.1.1/AC The TSF shall enforce the Access Control SFP[207] to transmit[208] user data in a manner protected from unauthorised disclosure.

Application note 38: The SFR FDP_UCT.1/AC requires the ability to encrypt the command data in a TPM command.

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

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

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit another trusted IT product[209] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for
(1) an authorisation session,
(2) an encryption session, identified by the encrypt or decrypt attribute of the session
in order to transfer commands and responses between the other trusted IT product and the TOE.[210]

Application note 39: An authorisation session or an encryption session is established by the command TPM2_StartAuthSession. The integrity protection of an authorisation session shall be implemented using a HMAC digest over the command or response data including the parameters as defined in [7]. In an encrypted session, only the first parameter shall be encrypted as long as the parameter has a size field [7].

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

[207] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[208] [selection: *transmit, receive*]
[209] [selection: *the TSF, another trusted IT product*]
[210] [assignment: *list of functions for which a trusted channel is required*]

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

FMT_MOF.1.1/AC The TSF shall restrict the ability to disable and enable[211] the functions TPM2_Clear[212] to Platform firmware and the lockout administrator[213].

7.1.5.2 Non-Volatile Storage

The non-volatile memory (NV memory) is used to memorise values across power events. Especially the following values are stored in the NV memory:

- NV index values,
- objects in the TPM object hierarchy that were made persistent using the TPM2_EvictControl command (see section 37.3 of [7]),
- saved operational variables by TPM2_Shutdown(TPM_SU_STATE) as addressed in Table 9 and the corresponding SFRs,
- persistent NV data as defined in section 37.5 of [7].

NV index values may be implemented as hybrid indices in order to maintain high frequency updates. In that case the values are held in the TPM RAM as well as in the NV memory. The update is processed on the values in RAM. On index-type dependent events the values in NV memory are synchronised with the values in RAM (see section 37.2.4 of [7]).

Application note 40: The TPM library specification allows usage of an external device for storing non-volatile NV data (see section 37.7.2 of [7]). If this option will be implemented, the ST writer shall model this inter-TSF user data transfer by additional SFRs FDP_UCT.1 and FDP_UIT.1.

FDP_ACC.1/NVM Subset access control (non-volatile memory)
Hierarchical to: No other components.
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/NVM The TSF shall enforce the NVM SFP[214] on
Subjects:
(1) Platform firmware,
(2) Platform owner,
(3) USER,
(4) ADMIN,
(5) World
Objects:
(1) (ordinary, counter, bit field, extended, pin pass, pin fail) NV index,
(2) objects of the TPM hierarchy
Operations:

[211] [selection: *determine the behavior of, disable, enable, modify the behaviour of*]
[212] [assignment: *list of functions*]
[213] [assignment: *the authorised identified roles*]
[214] [assignment: *access control SFP*]

(1) TPM2_NV_DefineSpace
(2) TPM2_NV_UndefineSpace
(3) TPM2_NV_UndefineSpaceSpecial
(4) TPM2_NV_Read
(5) TPM2_NV_ReadPublic
(6) TPM2_NV_Increment
(7) TPM2_NV_Extend
(8) TPM2_NV_SetBits
(9) TPM2_NV_Write
(10) TPM2_NV_ReadLock
(11) TPM2_NV_WriteLock
(12) TPM2_NV_Certify
(13) TPM2_EvictControl[215].

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

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

FDP_ACF.1.1/NVM The TSF shall enforce the NVM SFP[216] to objects based on the following:
Subjects as defined in Table 7:
(1) Platform firmware, Platform owner, USER, ADMIN, World with the security attributes
  (a) authentication status,
  (b) physical presence if supported by the TOE
Objects as defined in Table 8:
(1) NV index, NV counter index, NV bit field index, NV extend index, NV pin pass index, NV pin fail index with the security attributes:
  (a) NV attributes,
  (b) status whether physical presence is required for Platform firmware authorisation[217]

FDP_ACF.1.2/NVM The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
(1) The Platform firmware authenticated with platformAuth, platformPolicy or physical presence if supported by the TOE and the Platform owner are authorised to reserve space to hold the data associated with that index (TPM2_NV_DefineSpace). The physical presence is not required if it is not supported by the TOE or disabled for TPM2_NV_DefineSpace command.
(2) The Platform firmware authenticated with platformAuth, platformPolicy or physical presence if supported by the TOE and the Platform owner are authorised to remove a NV index (TPM2_NV_UndefineSpace). The physical presence is not required if it is not supported by the TOE or disabled for TPM2_NV_UndefineSpace command.

---

[215] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[216] [assignment: *access control SFP*]
[217] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(15) An authorised subject is allow to certify the contents of an NV index or a portion of an NV index using the command TPM2_NV_Certify[218]

FDP_ACF.1.3/NVM The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none[219].

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

(1) If phEnableNV is CLEAR

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

b) The platform cannot define (TPM_RC_HIERARCHY) or undefined (TPM_RC_HANDLE) indices[220].

Application note 41: The blocking of read or write access to NV indices shall be reset on TPM Reset or TPM Restart. This is addressed in the TPM state control SFP, see FDP_ACF.1/States and Table 9.

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

[218] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[219] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[220] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

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

FMT_MSA.1.1/NVM TSF shall enforce the NVM SFP[221] to restrict the ability to query and modify[222] the security attributes NV index attributes[223] to the authorised role of the subject that executes the NV related command.[224]

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

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

FMT_MSA.3.1/NVM The TSF shall enforce the NVM SFP[225] to provide restrictive[226] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/NVM The TSF shall allow ~~the~~ nobody[227] to specify alternative initial values to override the default values when an object or information is created.

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

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

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

(1) If TPMA_NV_READ_STCLEAR of the NV Index is SET and the *authPolicy* of the NV Index is provided and
   a) TPMA_NV_PPREAD is set and *platformAuth* is provided or
   b) TPMA_NV_OWNERREAD is set and *ownerAuth* is provided or
   c) TPMA_NV_AUTHREAD is set and *authValue* is provided

   the command TPM2_NV_ReadLock shall SET TPMA_NV_READLOCKED for the NV Index. TPMA_NV_READLOCKED will be CLEAR by the next TPM2_Startup(TPM_SU_CLEAR).

(2) If TPMA_NV_WRITEDEFINE or TPMA_NV_WRITE_STCLEAR attributes of an NV location are SET and the *authPolicy* of the NV Index is provided and
   a) TPMA_NV_PPWRITE is set and *platformAuth* is provided or
   b) TPMA_NV_OWNERWRITE is set and *ownerAuth* is provided or
   c) TPMA_NV_AUTHWRITE is set and *authValue* is provided

[221] [assignment: *access control SFP(s), information flow control SFP(s)*]
[222] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[223] [assignment: *list of security attributes*]
[224] assignment: *the authorised identified roles*]
[225] [assignment: *access control SFP, information flow control SFP*]
[226] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[227] [assignment: *the authorised identified roles*]

the command TPM2_NV_WriteLock shall SET TPMA_NV_WRITELOCKED for the NV Index. TPMA_NV_WRITELOCKED will be clear on the next TPM2_Startup(TPM_SU_CLEAR) unless TPMA_NV_WRITEDEFINE is SET.

Application note 42: If TPMA_NV_READ_STCLEAR of the NV Index is CLEAR, then the TPM shall return on command TPM2_NV_ReadLock the TPM_RC_NV_ATTRIBUTE. If neither TPMA_NV_WRITEDEFINE nor TPMA_NV_WRITE_STCLEAR of the NV Index is SET, then the TPM shall return on command TPM2_NV_WriteLock the TPM_RC_ATTRIBUTES.

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

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

FMT_MTD.1.1/NVM The TSF shall restrict the ability to modify[228] the authorisation secret (authValue) for a NV index[229] to ADMIN[230] using the command TPM2_NV_ChangeAuth.

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

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

FDP_ITC.1.1/NVM The TSF shall enforce the NVM SFP[231] when importing user data, controlled under the SFP, from outside of the TOE.

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

FDP_ITC.1.3/NVM The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: none[232]

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

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

FDP_ETC.1.1/NVM The TSF shall enforce the NVM SFP[233] when exporting user data, controlled under the SFP(s), outside of the TOE.

---

[228] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[229] [assignment: *list of TSF data*]
[230] [assignment: *the authorised identified roles*]
[231] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[232] [assignment: *additional importation control rules*]
[233] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

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

7.1.5.3 Credentials

Credentials in the context of this PP are understood as a means to provide evidence that a dedicated TPM object is resident on an authentic TPM. To get a credential, the protocol involves two parties: the initiator of the credential process and the credential provider. On request by the initiator the credential provider shall generate the evidence. For privacy reasons, the generated credential shall not contain the identity of a particular TPM where the object belongs to. Instead, the credential shall prove that the object is resident on a TPM that the credential provider believes to be authentic.

FDP_ACC.1/Cre Subset access control (credentials)

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Cre The TSF shall enforce the Credential SFP[234] on

Subjects

(1) USER,

(2) ADMIN,

(3) World

Objects

(1) Credential

Operations

(1) TPM2_ActivateCredential.[235]

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

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/Cre The TSF shall enforce the Credential SFP[236] to objects based on the following:

Subjects

(1) USER with authentication state gained with userAuth or authPolicy,

(2) ADMIN with authentication state gained with adminAuth or authPolicy,

(3) World with no security attributes

Objects

(1) Credential with security attribute HMAC over the credential BLOB[237].

---

[234] [assignment: *access control SFP*]

[235] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[236] [assignment: *access control SFP*]

[237] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

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

(1) The subject World is authorised to create a credential using the command TPM2_MakeCredential.

(2) The subject of role ADMIN regarding the object for which the credential was created and the role USER regarding the key for the decryption of the credential BLOB is authorised to activate the credential using the command TPM2_ActivateCredential[238].

FDP_ACF.1.3/Cre The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none[239].

FDP_ACF.1.4/Cre The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[240].

FMT_MSA.3/Cre Static attribute initialisation (credentials)

Hierarchical to: No other components.

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

FMT_MSA.3.1/Cre The TSF shall enforce the Credential SFP[241] to provide restrictive[242] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Cre The TSF shall allow ~~the~~ nobody[243] to specify alternative initial values to override the default values when an object or information is created.

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

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/Cre TSF shall enforce the Credential SFP[244] to restrict the ability to use[245] the security attributes HMAC in the credential BLOB[246] to USER[247].

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

---

238 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

239 [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

240 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

241 [assignment: *access control SFP, information flow control SFP*]

242 [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

243 [assignment: *the authorised identified roles*]

244 [assignment: *access control SFP(s), information flow control SFP(s)*]

245 [selection: *change_default, query, modify, delete, [assignment: other operations]*]

246 [assignment: *list of security attributes*]

247 [assignment: *the authorised identified roles*]

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

FCO_NRO.1.1/Cre The TSF shall be able to generate evidence of origin for transmitted TPM objects[248] at the request of the originator[249].

FCO_NRO.1.2/Cre The TSF shall be able to relate the information whether the object is resident in an authentic TPM[250] of the originator of the information, and the name and the public area of the TPM object[251] of the information to which the evidence applies.

FCO_NRO.1.3/Cre The TSF shall provide a capability to verify the evidence of origin of information to the initiator[252] given based on a credential BLOB that was generated by the credential provider[253].

7.2 Security assurance requirements

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

Table 10: Security assurance requirements for the TOE

Assurance Class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_FLR.1 Basic flow remediation

[248] [assignment: *list of information types*]
[249] [selection: *originator, recipient, [assignment: list of third parties]*]
[250] [assignment: *list of attributes*]
[251] [assignment: *list of information fields*]
[252] [selection: *originator, recipient, [assignment: list of third parties]*]
[253] [assignment: *limitations on the evidence of origin*]

Assurance Class	Assurance components
	ALC_TAT.1 Well-defined development tools
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: security enforcing modules
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.4 Methodical vulnerability analysis

2567

7.3 Security Requirements rationale

7.3.1 Sufficiency of SFR

The following table demonstrates that each security objective for the TOE is covered by at least one SFR and each SFR is traced back to at least one security objective for the TOE.

Table 11: Security requirements rationale

	O.Context_Management	O.Crypto_Key_Man	O.ECDAA	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl
FMT_SMR.1																	x					x
FMT_SMF.1																x						
FMT_MSA.2																x						
FCS_RNG.1		x																				
FPT_STM.1													x									
FIA_SOS.2								x														
FMT_MTD.1/AUTH								x														
FIA_AFL.1/Lockout								x											x			
FIA_AFL.1/Recover								x											x			
FIA_AFL.1/PINFAIL								x											x			
FIA_AFL.1/PINPASS								x											x			
FIA_AFL.1**/PINPASS Authentication failure handling** Hierarc								x		x												

	O.Context_Management	O.Crypto_Key_Man	O.ECDAA	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl
Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINPASS The TSF shall detect when pinCount successful authentication events exceeds pinLimit for an																						

		O.Context_Management	O.Crypto_Key_Man	O.ECDAA	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl
	NV Index with the attribute TPM_NT_PIN_PASS.																						
FIA_AFL.1.2/ PINPASS	When the defined number of successful authentication events has been met, the TSF shall block further authorization attempts.																						
FIA_AFL.1/PINFAIL Authentication failure handling	Hierarc																						

	O.Context_Management	O.Crypto_Key_Man	O.ECDAA	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl
Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINFAIL The TSF shall detect when																						

		O.Context_Management	O.Crypto_Key_Man	O.ECDAA	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl
	pinCount unsuccessful authentication attempts exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_FAIL.																						
FIA_AFL.1.2/ PINFAIL	When the defined number of unsuccessful authentication attempts has been met, the TSF shall block further																						

	O.Context_Management	O.Crypto_Key_Man	O.ECDAA	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl
authorization attempts. **FIA_UID.1**																						
FIA_UAU.1								x		x												
FIA_UAU.5								x			x	x								x		x
FIA_UAU.6								x											x			
FIA_USB.1				x				x			x						x					
FMT_MSA.4/AUTH				x				x								x						
FDP_ACC.2/States				x																		x
FDP_ACF.1/States				x																		x
FMT_MSA.1/States				x												x						x
FMT_MSA.3/States				x												x						x
FDP_UIT.1/States									x													x
FPT_TST.1						x	x											x				
FDP_ACC.1/AC				x							x											
FDP_ACF.1/AC				x							x											
FMT_MSA.1/AC				x							x					x						
FMT_MSA.3/AC				x							x					x						
FDP_UCT.1/AC																				x		
FTP_ITC.1/AC																				x		
FMT_MOF.1/AC				x																		
FCS_CKM.1/PK		x																				
FCS_CKM.1/ECC		x	x		x			x														
FCS_CKM.1/RSA		x																				
FCS_CKM.1/SYMM		x																				
FCS_CKM.4		x																				
FCS_COP.1/AES	x	x			x				x											x		
FCS_COP.1/SHA												x	x							x		
FCS_COP.1/HMAC	x	x			x			x	x											x		

	O.Context_Management	O.Crypto_Key_Man	O.ECDAA	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl
FCS_COP.1/RSAED					x				x											x		
FCS_COP.1/RSASign													x		x							
FCS_COP.1/ECDSA													x		x							
FCS_COP.1/ECDAA			x																			
FCS_COP.1/ECDEC					x			x														
FDP_ACC.1/NVM				x																		
FDP_ACF.1/NVM				x																		
FMT_MSA.1/NVM				x												x						
FMT_MSA.3/NVM				x												x						
FMT_MSA.4/NVM				x												x						
FMT_MTD.1/NVM				x												x						
FDP_ITC.1/NVM									x													
FDP_ETC.1/NVM					x																	
FDP_ACC.1/ExIm				x	x				x													
FDP_ACF.1/ExIm				x	x				x													
FMT_MSA.1/ExIm				x	x				x							x						
FMT_MSA.3/ExIm				x	x				x							x						
FDP_ETC.2/ExIm	x				x																	
FDP_ITC.2/ExIm	x								x													
FDP_UCT.1/ExIm	x				x				x													
FDP_UIT.1/ExIm	x				x				x				x									
FDP_ACC.1/Cre													x									
FDP_ACF.1/Cre													x									
FMT_MSA.1/Cre													x			x						
FMT_MSA.3/Cre													x			x						
FCO_NRO.1/Cre							x						x		x							
FDP_ACC.1/M&R				x								x										
FDP_ACF.1/M&R			x	x								x										
FMT_MSA.1/M&R				x								x			x	x						
FMT_MSA.3/M&R				x								x			x	x						
FCO_NRO.1/M&R			x										x		x							

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

A detailed justification required for suitability of the security functional requirements to achieve the security objectives is given below.

The security objective **O.Context_Management** requires that the TOE protects the confidentiality and integrity of the data of a resource and allows the restoring of the resource on the same TPM and during the same operational cycle only. This objective is addressed by the following SFRs:

- FDP_ETC.2/ExIm requires that the TSF shall apply a policy when exporting user data. The policy rules require the protection of data integrity and confidentiality at export of objects from the TPM hierarchy.
- FDP_ITC.2/ExIm require that the TSF shall apply a policy when importing user data. The security attributes shall be unambiguously associated with the user data while importing.
- FDP_UCT.1/ExIm require that the TSF protects the confidentiality of transmitted user data while data export and import.
- FDP_UIT.1/ExIm require that the TSF protects the integrity of transmitted user data while data export and import.
- FCS_COP.1/HMAC requires that the TSF provides the ability to generate and verify HMAC values.
- FCS_COP.1/AES requires that the TSF provides the ability to perform symmetric encryption and decryption of data.

The security objective **O.Crypto_Key_Man** requires the secure management of cryptographic keys including its generation using the TOE random number generator as source of randomness. This objective is addressed by the following SFRs:

- FCS_CKM.1/PK requires the TSF to generate primary cryptographic keys by means of defined key generation functions.
- FCS_CKM.1/RSA requires the TSF to generate cryptographic RSA keys in accordance with a assigned key generation algorithm.
- FCS_CKM.1/ECC requires the TSF to generate cryptographic ECC keys in accordance with a assigned key generation algorithm.
- FCS_CKM.1/SYMM requires the TSF to generate cryptographic symmetric keys in accordance with a assigned key generation algorithm.
- FCS_CKM.4 requires the TSF to be able destroy cryptographic keys in accordance with a specific key destruction method.
- FCS_COP.1/HMAC requires that the TSF provides the ability to generate and verify HMAC values. This is required for the specified key generation algorithm according to FCS_CKM.1/PK.
- FCS_COP.1/AES requires that the TSF provides the ability to perform symmetric encryption and decryption of data.
- FCS_RNG.1 requires the TSF to provide a random number generator. This is used as source of randomness in the specified key generation algorithm according to FCS_CKM.1/PK.

The security objective **O.ECDAA** requires that the TOE to implement the TPM part of the ECDAA signing operation. This is directly addressed by the SFRs FCS_CKM.1/ECC and FCS_COP.1/ECDAA: While FCS_CKM.1/ECC requires the ability of the TSF to generate ECC keys, the SFR FCS_COP.1/ECDAA requires the TSF to implement the ECDAA algorithm, and to the proof of origin FDP_ACF.1/M&R and FCO_NRO.1/M&R.

The security objective **O.DAC** requires that the TOE controls and restricts user access to the TOE protected capabilities and shielded locations in accordance with the specified access control policies. The object owner shall manage the access rights using the principle of least privilege. This objective is addressed by the following SFRs:

- FIA_USB.1 addresses the association between subjects and its security attributes. The SFR defines rules for initial association and changes of these associations.
- FDP_ACC.2/States requires that the TSF enforces the TPM State Control SFP on all subjects, objects and operations among subjects and objects covered by the SFP. The operations shall be covered by an access control SFP.
- FDP_ACF.1/States defines rules to enforce a policy regarding the TOE states, transitions between states and required authorisations to change the state of the TOE.
- FMT_MSA.1/States requires that a TSF shall enforce the TPM State Control SFP to restrict the ability to modify the TOE state.
- FMT_MSA.3/States requires that the TSF shall enforce the TPM State Control SFP to provide restrictive default values for security attributes and nobody is authorised to specify alternative default initial values.

- FDP_ACC.1/AC requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects and operations among subjects and objects covered by the SFP.
- FDP_ACF.1/AC defines rules to enforce a policy regarding the TOE access control and the required authorisations to perform dedicated operations.
- FMT_MSA.1/AC requires that a TSF shall enforce a SFP to restrict the ability to perform dedicated operations on security attributes to dedicated authorised roles.
- FMT_MSA.3/AC requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes and only dedicated roles are authorised to specify alternative default initial values.
- FMT_MSA.4/AUTH defines rules to disable the security attributes authValue and authPolicy.
- FMT_MOF.1/AC requires the TSF to restrict the ability to disable and enable the TPM2_Clear function to dedicated authorised roles.
- FDP_ACC.1/NVM requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects and NVM related operations among subjects and objects covered by the SFP.
- FDP_ACF.1/NVM defines rules to enforce a policy regarding the NVM access control and the required authorisations to perform dedicated operations.
- FMT_MSA.1/NVM requires that a TSF shall enforce a SFP to restrict the ability to query and modify NV index attributes.
- FMT_MSA.3/NVM requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes and nobody is authorised to specify alternative default initial values.
- FMT_MSA.4/NVM requires management of security attributes controlling read access to NVM.
- FMT_MTD.1/NVM requires that the TSF restricts the ability to change the authorisation secret for an NV index to a special role.
- FDP_ACC.1/ExIm requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects and export/import operations among subjects and objects covered by the SFP.
- FDP_ACF.1/ExIm defines rules to enforce a policy regarding the access control and the required authorisations to perform export and import related operations.
- FMT_MSA.1/ExIm requires that a TSF shall enforce a SFP to restrict the ability to use the security attribute *authorisation data* for export and import related functions.
- FMT_MSA.3/ExIm requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes for export and import related functions. Nobody is authorised to specify alternative default initial values for those security attributes.
- FDP_ACC.1/M&R requires that the TSF enforces a SPF for access control regarding dedicated subjects, PCR and corresponding operations covered by the SFP.
- FDP_ACF.1/M&R defines rules to enforce a policy regarding the PCR access control and the required authorisations to perform PCR related operations.
- FMT_MSA.1/M&R requires that a TSF shall enforce a SFP to restrict the ability to modify the PCR related security attributes.

- FMT_MSA.3/M&R requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes for measurement and reporting related functions. Nobody is authorised to specify alternative default initial values for those security attributes.
- FDP_ACC.1/Hier requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects and TPM hierarchy related operations among subjects and objects covered by the SFP.
- FDP_ACF.1/Hier defines rules to enforce a policy regarding the access control and the required authorisations to perform TPM hierarchy related operations.
- FMT_MSA.1/Hier requires that a TSF shall enforce a SFP to restrict the ability to modify the security attributes fixedTPM and fixedParent.
- FMT_MSA.3/Hier requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes for TPM hierarchy related operations. The creator of the TPM object is authorised to specify alternative default initial values for those security attributes.
- FMT_MSA.4/Hier limits the management of security attributes of hierarchies.

The security objective **O.Export** requires that the TOE protects the confidentiality and integrity of data in case of export. Further, the TOE shall unambiguously associate the data security attributes with the data to be exported. This objective is addressed by the following SFRs:

- FCS_COP.1/RSAED requires that the TSF provides the ability to perform RSA based asymmetric encryption and decryption of data.
- FCS_COP.1/ECDEC requires that the TSF provides the ability to perform elliptic curve based asymmetric decryption of data.
- FCS_CKM.1/ECC requires that the TSF provides the ability to generate keys for elliptic curve based algorithms.
- FCS_COP.1/AES requires that the TSF provides the ability to perform symmetric encryption and decryption of data.
- FCS_COP.1/HMAC requires that the TSF provides the ability to generate and verify HMAC values.
- FDP_ETC.1/NVM requires that the TSF enforces a SFP when exporting user data from NV memory.
- FDP_ACC.1/ExIm requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects and export/import operations among subjects and objects covered by the SFP.
- FDP_ACF.1/ExIm defines rules to enforce a policy regarding the access control and the required authorisations to perform export and import related operations.
- FMT_MSA.1/ExIm requires that a TSF shall enforce a SFP to restrict the ability to use the security attribute *authorisation data* for export and import related functions.
- FMT_MSA.3/ExIm requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes for export and import related functions. Nobody is authorised to specify alternative default initial values for those security attributes.

- FDP_ETC.2/ExIm requires that the TSF shall apply a policy when exporting user data. The policy rules require the protection of data integrity and confidentiality at export of objects from the TPM hierarchy.
- FDP_UCT.1/ExIm require that the TSF protects the confidentiality of transmitted user data while data export and import.
- FDP_UIT.1/ExIm require that the TSF protects the integrity of transmitted user data while data export and import.
- FDP_SDI.1 requires that the TSF shall monitor stored user data for modification and modification of the TPM hierarchies.

The security objective **O.Fail_Secure** requires that the TOE enters a secure failure mode in case of a failure. To address this security objective, FPT_FLS.1/FS requires the TSF to preserve a secure state by entering a fail state and FPT_FLS.1/SD requires the TSF to preserve a secure state by shutdown of the TOE. FPT_TST.1 requires the TSF to provide self tests in order to detect failure situations.

The security objective **O.General_Integ_Checks** requires the ability of the TOE to check the system integrity and user data integrity. This objective is addressed by the following SFRs:

- FPT_TST.1 requires the TSF to provide self tests in order to detect failure situations. This self tests may include tests of the system and data integrity.
- FCO_NRO.1/Cre requires the TSF to be able to generate evidence of origin for transmitted TPM objects and to verify this evidence.
- FDP_SDI.1 requires that the TSF shall monitor stored user data for modification and modification of the TPM hierarchies.

The security objective **O.I&A** requires that the TOE identifies all users and authenticates the claimed identity except the role "World" before granting a user access to the TOE facilities. This objective is addressed by the following SFRs:

- FIA_SOS.2 requires the TSF to generate secrets for usage in the authentication functionality.
- FMT_MTD.1/AUTH requires the TSF to restrict the management of authentication data to dedicated authorised roles.
- FMT_MSA.4/AUTH defines rules for management of the security attributes controlling the use of authentication mechanisms for authorisation of objects.
- FIA_AFL.1/Lockout, FIA_AFL.1/Recover, FIA_AFL.1/PINPASS and FIA_AFL.1/PINFAIL require the TSF to detect attacks to the authentication system by a number of ongoing unsuccessful authentication requests. On detection the TSF shall block that authentication method.
- FIA_AFL.1**/PINPASS Authentication failure handling**
  Hierarchical to: No other components.
  Dependencies: FIA_UAU.1 Timing of authentication.

FIA_AFL.1.1/PINPASS The TSF shall detect when pinCount successful authentication events exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_PASS.
FIA_AFL.1.2/ PINPASS When the defined number of successful authentication events has been met, the TSF shall block further authorization attempts.

FIA_AFL.1/PINFAIL Authentication failure handling

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

FIA_AFL.1.1/PINFAIL The TSF shall detect when pinCount unsuccessful authentication attempts exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_FAIL.

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

- **FIA_UID.1** requires the TSF to allow dedicated commands before an user is identified. For any other TSF mediated action the TSF shall require the successful identification of the user.
- FIA_UAU.1 requires the TSF to allow dedicated commands before an user is authenticated. For any other TSF mediated action the TSF shall require the successful authentication of the user.
- FIA_UAU.5 requires the TSF to provide dedicated authentication mechanisms. Further, the TSF shall follow the given rules when authenticating any user's identity.
- FIA_UAU.6 requires the TSF to re-authenticate the user when multiple commands need to be executed in one authorisation session.
- FIA_USB.1 addresses the association between subjects and its security attributes. The SFR defines rules for initial association and changes of these associations.
- FCS_COP.1/HMAC requires that the TSF provides the ability to generate and verify HMAC values. This is used for integrity and authenticity verification.
- FCS_CKM.1/ECC requires that the TSF provides the ability to generate keys for elliptic curve based algorithms.
- FCS_COP.1/ECDEC requires that the TSF provides the ability to perform elliptic curve based asymmetric decryption of data.

The security objective **O.Import** requires that the TOE ensures that the data security attributes are being imported with the imported data and that the data is from authorised source. Further, the TOE shall verify the security attributes according to the TSF access control rules. The TOE shall support the protection of confidentiality and the verification of the integrity of imported data (except the verification of the integrity of the data within a sealed data blob). This objective is addressed by the following SFRs:

- FDP_UIT.1/States requires that the TSF shall enforce a SFP to provide and use integrity protection capabilities for firmware update data on reception of that data.
- FCS_COP.1/RSAED requires that the TSF provides the ability to perform asymmetric encryption and decryption of data.
- FCS_COP.1/HMAC requires that the TSF provides the ability to generate and verify HMAC values.
- FCS_COP.1/AES requires that the TSF provides the ability to perform symmetric encryption and decryption of data.
- FDP_ITC.1/NVM requires that the TSF enforces a SFP when importing user data controlled under the SFP. The TSF shall enforce given rules on import of those user data.

- FDP_ACC.1/ExIm requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects and export/import operations among subjects and objects covered by the SFP.
- FDP_ACF.1/ExIm defines rules to enforce a policy regarding the access control and the required authorisations to perform export and import related operations.
- FMT_MSA.1/ExIm requires that a TSF shall enforce a SFP to restrict the ability to use the security attribute *authorisation data* for export and import related functions.
- FMT_MSA.3/ExIm requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes for export and import related functions. Nobody is authorised to specify alternative default initial values for those security attributes.
- FDP_ITC.2/ExIm require that the TSF shall apply a policy when importing user data. The security attributes shall be unambiguously associated with the user data while importing.
- FDP_UCT.1/ExIm require that the TSF protects the confidentiality of transmitted user data while data export and import.
- FDP_UIT.1/ExIm require that the TSF protects the integrity of transmitted user data while data export and import.
- FDP_SDI.1 requires that the TSF shall monitor stored user data for modification and modification of the TPM hierarchies.

The security objective **O.Limit_Actions_Auth** requires that the TOE restricts the actions a user may perform before the TOE verified the identity of the user. This includes requirements for physical presence of the platform firmware if physical presence is supported and enabled for the required command. This is directly addressed by the SFR FIA_UAU.1 which requires the TSF to allow only dedicated commands before an user is authenticated. For any other TSF mediated action the TSF shall require the successful authentication of the user.

The security objective **O.Locality** requires that the TOE controls the access to objects based on the locality of the process communicating with the TPM. This objective is addressed by the following SFRs:

- FIA_UAU.5 requires the TSF to provide dedicated authentication mechanisms. Further, the TSF shall follow the given rules when authenticating any user's identity.
- FDP_ACC.1/AC requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects and operations among subjects and objects covered by the SFP.
- FDP_ACF.1/AC defines rules to enforce a policy regarding the TOE access control and the required authorisations to perform dedicated operations.
- FMT_MSA.1/AC requires that a TSF shall enforce a SFP to restrict the ability to perform dedicated operations on security attributes to dedicated authorised roles.
- FMT_MSA.3/AC requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes and only dedicated roles are authorised to specify alternative default initial values.
- FIA_USB.1 addresses the association between subjects and its security attributes. The SFR defines rules for initial association and changes of these associations.

The security objective **O.Record_Measurement** requires that the TOE supports calculating hash values and recording the result of a measurement. This is directly addressed by the SFR FCS_COP.1/SHA which requires the TSF to be able to perform hash value calculations. The aspect of recording the results is realised by the ability of the TOE to derive access control measures based on the result of measurement. The SFRs FIA_UAU.5, FMT_MSA.1/M&R and FMT_MSA.3/M&R are involved in that ability: FIA_UAU.5 requires the TSF to provide dedicated authentication mechanisms including policy based authentication using the value of PCR. Further, the TSF shall follow the given rules when authenticating any user's identity. The SFR FMT_MSA.1/M&R requires that a TSF shall enforce a SFP to restrict the ability to modify the PCR related security attributes, FMT_MSA.3/M&R requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes for measurement and reporting related functions. Nobody is authorised to specify alternative default initial values for those security attributes. Regarding the access control of PCR related operations the following SFRs support the security objective:

- FDP_ACC.1/M&R requires that the TSF enforces a SPF for access control regarding dedicated subjects, PCR and corresponding operations covered by the SFP.
- FDP_ACF.1/M&R defines rules to enforce a policy regarding the PCR access control and the required authorisations to perform PCR related operations.

The security objective **O.MessageNR** requires that the TOE provides user data integrity, source authentication and the basis for source non-repudiation when exchanging data with a remote system. This objective is addressed by the following SFRs:

- FPT_STM.1 requires that the TSF is able to provide reliable timestamps.
- FCS_COP.1/SHA requires the TSF to be able to perform hash value calculations. This can be used to support data integrity protection.
- FCS_COP.1/RSASign requires the TSF to be able to perform signature generation and verification. This can be used to support source authentication and source non-repudiation when exchanging data with a remote system.
- FCS_COP.1/ECDSA requires the TSF to be able to perform signature generation and verification. This can be used to support source authentication and source non-repudiation when exchanging data with a remote system.
- FDP_UIT.1/ExIm require that the TSF protects the integrity of transmitted user data while data export and import.
- FDP_ACC.1/Cre requires that the TSF enforces a SPF for access control regarding the handling of credentials.
- FDP_ACF.1/Cre defines rules to enforce a policy regarding the handling of credentials.
- FMT_MSA.1/Cre requires that a TSF shall enforce a SFP to restrict the ability to manage credentials for TPM objects.
- FMT_MSA.3/Cre requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes and nobody is authorised to specify alternative default initial values.
- FCO_NRO.1/Cre requires the TSF to be able to generate evidence of origin for transmitted TPM objects and to verify this evidence.

- FCO_NRO.1/M&R requires the TSF to be able to generate evidence of origin for transmitted attestation structures and to verify this evidence.

The security objective **O.No_Residual_Info** requires that there is no residual information in information containers or system resources upon their reallocation to different users. This objective is directly addressed by the SFR FDP_RIP.1 that requires that the TSF ensures that any previous information content of any object is made unavailable upon the deallocation of the resource.

The security objective **O.Reporting** requires that the TOE reports measurement digests and attests to the authenticity of measurement digests. This objective is addressed by the following SFRs:

- FCS_COP.1/RSASign requires the TSF to be able to perform signature generation and verification. This can be used to support authentication of measurement digests.
- FCS_COP.1/ECDSA requires the TSF to be able to perform signature generation and verification. This can be used to support authentication of measurement digest.
- FCO_NRO.1/Cre requires the TSF to be able to generate evidence of origin for transmitted TPM objects and to verify this evidence.
- FMT_MSA.1/M&R requires that a TSF shall enforce a SFP to restrict the ability to modify the PCR related security attributes.
- FMT_MSA.3/M&R requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes for measurement and reporting related functions. Nobody is authorised to specify alternative default initial values for those security attributes.
- FCO_NRO.1/M&R requires the TSF to be able to generate evidence of origin for transmitted attestation structures and to verify this evidence.

The security objective **O.Security_Attr_Mgt** requires that the TOE allows only authorised users to initialise and to change security attributes of objects and subjects. This management shall be based on least privilege by means of role based administration and separation of duty. This objective is addressed by the following SFRs:

- FMT_SMF.1 requires the TSF to be able to perform different management functions which are listed in the SFR.
- FMT_MSA.2 requires that the TSF only accepts secure values for the security attributes that are listed in the SFR.
- FMT_MSA.4/AUTH defines rules to disable the security attributes authValue and authPolicy.
- FMT_MSA.1/States requires that a TSF shall enforce the TPM state control SFP to restrict the ability to modify the TOE state.
- FMT_MSA.3/States requires that the TSF shall enforce the TPM state control SFP to provide restrictive default values for security attributes and nobody is authorised to specify alternative default initial values.
- FMT_MSA.1/AC requires that a TSF shall enforce the Access Control SFP to restrict the ability to modify the TOE state.
- FMT_MSA.3/AC requires that the TSF shall enforce the Access Control SFP to provide restrictive default values for security attributes and only dedicated roles are authorised to specify alternative default initial values.

- FMT_MSA.1/NVM requires that a TSF shall enforce the NVM SFP to restrict the ability to query and modify NV index attributes.
- FMT_MSA.3/NVM requires that the TSF shall enforce the NVM SFP to provide restrictive default values for NVM related security attributes and nobody is authorised to specify alternative default initial values.
- FMT_MSA.4/NVM requires that the TSF shall enforce rules for setting the security attributes of NVM.
- FMT_MTD.1/NVM requires that the TSF restricts the ability to change the authorisation secret for an NV index to a special role.
- FMT_MSA.1/ExIm requires that a TSF shall enforce the Data Export and Import_SFP to restrict the ability to use the security attribute *authorisation data* for export and import related functions.
- FMT_MSA.3/ExIm requires that the TSF shall enforce the Data Export and Import SFP to provide restrictive default values for security attributes for export and import related functions. Nobody is authorised to specify alternative default initial values for those security attributes.
- FMT_MSA.1/Cre requires that a TSF shall enforce the Credential_SFP to restrict the ability to modify the PCR related security attributes.
- FMT_MSA.3/Cre requires that the TSF shall enforce the Credential_SFP to provide restrictive default values for security attributes for measurement and reporting related functions. Nobody is authorised to specify alternative default initial values for those security attributes.
- FMT_MSA.1/M&R requires that a TSF shall enforce the Measurement and Reporting SFP to restrict the ability to modify the PCR related security attributes.
- FMT_MSA.3/M&R requires that the TSF shall enforce the Measurement and Reporting_SFP to provide restrictive default values for security attributes for measurement and reporting related functions. Nobody is authorised to specify alternative default initial values for those security attributes.
- FMT_MSA.1/Hier requires that a TSF shall enforce the TPM Object Hierarchy SFP to restrict the ability to modify the security attributes fixedTPM and fixedParent.
- FMT_MSA.3/Hier requires that the TSF shall enforce the TPM Object Hierarchy SFP to provide restrictive default values for security attributes for TPM hierarchy related operations. The creator of the TPM object is authorised to specify alternative default initial values for those security attributes.
- FMT_MSA.4/Hier requires that the TSF shall enforce rules for setting the security attributes of TPM object hierarchies.

The security objective **O.Security_Roles** requires that the TOE maintains security relevant roles and associates users with those roles. The SFR FMT_SMR.1 defines a set of roles that the TSF shall maintain. Also, the association of users with these roles is required by this SFR. Further, FIA_USB.1 addresses the association between subjects and its security attributes. The SFR defines rules for initial association and changes of these associations.

The security objective **O.Self_Test** requires the TOE to provide the ability to test itself and verify the integrity of the shielded data objects. Further, protected capabilities shall operate as designed and enter a secure state in case of detected errors. This is directly addressed by the SFRs FPT_TST.1, FPT_FLS.1/SD and FPT_FLS.1/FS:

- FPT_TST.1 requires the TSF to run self tests under special conditions that are defined in the SFR.
- FPT_FLS.1/FS requires the TSF to preserve a secure state when failures occur. The types of failures are given in the SFR.
- FPT_FLS.1/SD requires the TSF to preserve a safe state by shutdown of the TOE in case of a detected physical attack or when the environmental conditions are out of spec.

The security objective **O.Single_Auth** requires that the TOE provides a single user authentication mechanism. To prevent "replay" and "man-in-the-middle" attacks the TOE shall require re-authentication. This objective is addressed by the following SFRs:

- FIA_AFL.1/Lockout FIA_AFL.1/Recover, FIA_AFL.1/PINPASS and FIA_AFL.1/PINFAIL require the TSF to detect attacks to the authentication system by a number of ongoing unsuccessful authentication requests. On detection the TSF shall block that authentication method.
- FIA_UAU.6 requires the TSF to re-authenticate the user when multiple commands need to be executed in one authorisation session.

The security objective **O.Sessions** requires that the TOE provides the confidentiality of the parameters of commands within an authorised session and the integrity of the audit log of commands. This objective is addressed by the following SFRs:

- FIA_UAU.5 requires the TSF to provide dedicated authentication mechanisms. Further, the TSF shall follow the given rules when authenticating any user's identity. The given authentication mechanisms are used as basis for the establishment of the integrity and confidentiality protected communication channels.
- FDP_UCT.1/AC requires the TSF to enforce a policy to transmit user data in a confidential manner.
- FTP_ITC.1/AC requires that the TSF shall provide a communication channel between itself and the user of the TOE in a manner that protects the confidentiality and integrity of the transmitted data. This channel is used by authorisation sessions, audit sessions and encryption sessions of the TPM and used to transfer commands and responses between the TOE and the user of the TOE.
- FCS_COP.1/RSAED requires that the TSF provides the ability to perform asymmetric encryption and decryption of data.
- FCS_COP.1/SHA requires the TSF to be able to perform hash value calculations. This can be used to support data integrity protection.
- FCS_COP.1/HMAC requires that the TSF provides the ability to generate and verify HMAC values.
- FCS_COP.1/AES requires that the TSF provides the ability to perform symmetric encryption and decryption of data.

The security objective **O.Tamper_Resistance** requires the TOE to resist physical tampering of the TSF by hostile users. This security objective is directly addressed by the SFR FPT_PHP.3 which requires that the TSF resists physical manipulation and physical probing but also by the SFRs FDP_ITT.1 and FPT_ITT.1 that require the TSF to prevent the disclosure of user data when transmitted between physically separated parts of the TOE.

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

- FMT_SMR.1 defines a set of roles that the TSF shall maintain. Also, the association of users with these roles is required by this SFR.
- FDP_ACC.2/States requires that the TSF enforces the TPM State Control SFP on all subjects, objects and operations among subjects and objects covered by the SFP. The operations shall be covered by an access control SFP.
- FDP_ACF.1/States defines rules to enforce a policy regarding the TOE states, transitions between states and required authorisations to change the state of the TOE. This includes the state transition regarding the FUM state and the rules for the required authorisations.
- FMT_MSA.1/States requires that a TSF shall enforce a SFP to restrict the ability to modify the TOE state.
- FMT_MSA.3/States requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes and nobody is authorised to specify alternative default initial values.
- FDP_UIT.1/States requires that the TSF shall enforce the TPM State Control SFP to provide and use integrity protection capabilities for firmware update data on reception of that data.
- FIA_UAU.5: requires the TSF to provide dedicated authentication mechanisms. Further, the TSF shall follow the given rules when authenticating any user's identity.

7.3.2 Dependency Rationale

The dependency rationale demonstrates that the dependencies of the SFR are fulfilled or provides an explanation in case that dependencies are not fulfilled.

Table 12: SFR Dependency rationale

SFR	Dependency	Rationale/ fulfilled by
FMT_SMR.1	FIA_UID.1 Timing of identification	Fulfilled by FIA_AFL.1**/PINPASS Authentication failure handling** Hierarchic

SFR	Dependency	Rationale/ fulfilled by
		Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINPASS The TSF shall detect when <u>pinCount</u> successful authentication events exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_PASS. FIA_AFL.1.2/ PINPASS When the defined number of successful authentication events has been

SFR	Dependency	Rationale/ fulfilled by
		met, the TSF shall block further authorization attempts. **FIA_AFL.1/PINFAIL Authentication failure handling** Hierarchic Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINFAIL The TSF shall detect when

SFR	Dependency	Rationale/ fulfilled by
		pinCount unsuccessful authentication attempts exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_FAIL. FIA_AFL.1.2/ PINFAIL When the defined number of unsuccessful authentication attempts has been met, the TSF shall block further authorization attempts. **FIA_UID.1**
FMT_SMF.1	No dependencies	n. a.
FMT_MSA.2	[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	Fulfilled by FDP_ACC.1/AC, FMT_MSA.1/AC, FMT_SMR.1
FMT_MSA.4/AUTH	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/Hier,
FCS_RNG.1	No dependencies	n. a.
FPT_STM.1	No dependencies	n. a.
FIA_SOS.2	No dependencies	n. a.
FMT_MTD.1/AUTH	FMT_SMR.1 Security roles	Fulfilled by

SFR	Dependency	Rationale/ fulfilled by
	FMT_SMF.1 Specification of Management Functions	FMT_SMR.1, FMT_SMF.1
FIA_AFL.1/Lockout	FIA_UAU.1 Timing of authentication	Fulfilled by FIA_UAU.1
FIA_AFL.1/Recover	FIA_UAU.1 Timing of authentication	Fulfilled by FIA_UAU.1
FIA_AFL.1/PINFAIL	FIA_UAU.1 Timing of authentication	Fulfilled by FIA_UAU.1
FIA_AFL.1/PINPASS	FIA_UAU.1 Timing of authentication	Fulfilled by FIA_UAU.1
FIA_AFL.1**/PINPASS Authentication failure handling** Hierarchic Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINPASS The TSF shall detect when pinCount	No dependencies	n. a.

SFR	Dependency	Rationale/ fulfilled by
successful authentication events exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_PASS. FIA_AFL.1.2/ PINPASS When the defined number of successful authentication events has been met, the TSF shall block further authorization attempts. **FIA_AFL.1/PINFAIL Authentication failure handling** Hierarchic		

SFR	Dependency	Rationale/ fulfilled by
Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINFAIL The TSF shall detect when <u>pinCount</u> unsuccessful authentication attempts exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_FAIL. FIA_AFL.1.2/ PINFAIL When the defined number of unsuccessful authentication attempts has been met, the TSF shall block further authorizati		

SFR	Dependency	Rationale/ fulfilled by
on attempts. **FIA_UID.1**		
FIA_UAU.1	FIA_UID.1 Timing of identification	Fulfilled by FIA_AFL.1**/PINPASS Authentication failure handling** Hierarchic Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINPASS The TSF shall detect when pinCount successful authentic

SFR	Dependency	Rationale/ fulfilled by
		ation events exceeds pinLimit for an NV Index with the attribute TPM_NT_ PIN_PASS . FIA_AFL.1.2/ PINPASS Wh en the defined number of successfu l authentic ation events has been met, the TSF shall block further authorizat ion attempts. **FIA_AFL.1/PINFAIL Authentication failure handling** Hierarchic

SFR	Dependency	Rationale/ fulfilled by
		Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINFAIL The TSF shall detect when <u>pinCount</u> unsuccessful authentication attempts exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_FAIL. FIA_AFL.1.2/ PINFAIL When the defined number of unsuccessful authentication attempts has been met, the TSF shall block

SFR	Dependency	Rationale/ fulfilled by
		further authorization attempts. **FIA_UID.1**
FIA_UAU.5	No dependencies	n. a.
FIA_UAU.6	No dependencies	n. a.
FIA_USB.1	FIA_ATD.1 User attribute definition	Because the TOE does not identify or manage individual users, the SFR FIA_ATD.1 is not applicable here.
FDP_ACC.2/States	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/States
FDP_ACF.1/States	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.2/States, FMT_MSA.3/States
FMT_MSA.1/States	[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	Fulfilled by FDP_ACC.2/States, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/States	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/States, FMT_SMR.1
FDP_UIT.1/States	[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]	Fulfilled by FDP_ACC.2/States, see rationale (1) below this table
FPT_TST.1	No dependencies	n. a.
FDP_ACC.1/AC	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/AC
FDP_ACF.1/AC	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/AC, FMT_MSA.3/AC
FMT_MSA.1/AC	[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	Fulfilled by FDP_ACC.1/AC, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/AC	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/AC, FMT_SMR.1

SFR	Dependency	Rationale/ fulfilled by
FDP_UCT.1/AC	[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]	Fulfilled by FTP_ITC.1/AC, FDP_ACC.1/AC
FTP_ITC.1/AC	No dependencies	n. a.
FMT_MOF.1/AC	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	Fulfilled by FMT_SMR.1, FMT_SMF.1
FCS_CKM.1**/**PK	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/AES FCS_COP.1/RSAED, FCS_COP.1/RSASign FCS_CKM.4
FCS_CKM.1/RSA	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/RSAED, FCS_COP.1/RSASign FCS_CKM.4
FCS_CKM.1/ECC	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/ECDEC, FCS_COP.1/ECDSA, FCS_COP.1/ECDAA FCS_CKM.4
FCS_CKM.1/SYMM	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/AES FCS_CKM.4
FCS_CKM.4	[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]	Fulfilled by FCS_CKM.1**/**PK
FCS_COP.1/RSAED	[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	Fulfilled by FCS_CKM.1/PK, FCS_CKM.1/RSA, FCS_CKM.4
FCS_COP.1/SHA	[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	because hash functions do not use any keys, the dependencies regarding key generation/destruction are not applicable here
FCS_COP.1/HMAC	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with	Fulfilled by FCS_CKM.1/PK, FCS_CKM.1/SYMM,

SFR	Dependency	Rationale/ fulfilled by
	security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.4
FCS_COP.1/RSASign	[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	Fulfilled by FCS_CKM.1/PK, FCS_CKM.1/RSA FCS_CKM.4
FCS_COP.1/ECDSA	[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	Fulfilled by FCS_CKM.1/ECC, FCS_CKM.4
FCS_COP.1/ECDAA	[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	Fulfilled by FCS_CKM.1/ECC, FCS_CKM.4
FCS_COP.1/AES	[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	Fulfilled by FCS_CKM.1/PK, FCS_CKM.1/SYMM, FCS_CKM.4
FCS_COP.1/ECDEC	[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	Fulfilled by FCS_CKM.1/PK, FCS_CKM.1/ECC FCS_CKM.4
FDP_ACC.1/NVM	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/NVM
FDP_ACF.1/NVM	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/NVM, FMT_MSA.3/NVM
FMT_MSA.1/NVM	[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	Fulfilled by FDP_ACC.1/NVM, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/NVM	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/NVM, FMT_SMR.1

SFR	Dependency	Rationale/ fulfilled by
FMT_MSA.4/NVM	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/NVM
FMT_MTD.1/NVM	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled by FMT_SMR.1, FMT_SMF.1
FDP_ITC.1/NVM	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/NVM, FMT_MSA.3/NVM
FDP_ETC.1/NVM	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/NVM
FDP_ACC.1/ExIm	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/ExIm
FDP_ACF.1/ExIm	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/ExIm, FMT_MSA.3/ExIm
FMT_MSA.1/ExIm	[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	Fulfilled by FDP_ACC.1/ExIm, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/ExIm	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/ExIm, FMT_SMR.1
FDP_ETC.2/ExIm	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/ExIm
FDP_ITC.2/ExIm	[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	Fulfilled by FDP_ACC.1/ExIm, see rationale (2) and (3) below this table
FDP_UCT.1/ExIm	[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]	Fulfilled by FDP_ACC.1/ExIm, see rationale (3) below this table
FDP_UIT.1/ExIm	[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]	Fulfilled by FDP_ACC.1/ExIm, see rationale (3) below this table

SFR	Dependency	Rationale/ fulfilled by
FDP_ACC.1/Cre	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/Cre
FDP_ACF.1/Cre	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/Cre, FMT_MSA.3/Cre
FMT_MSA.3/Cre	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/Cre, FMT_SMR.1
FMT_MSA.1/Cre	[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	Fulfilled by FDP_ACC.1/Cre, FMT_SMR.1, FMT_SMF.1
FCO_NRO.1/Cre	FIA_UID.1 Timing of identification	Fulfilled by FIA_AFL.1**/PINPASS Authentication failure handling** Hierarchic Dependencies:

SFR	Dependency	Rationale/ fulfilled by
		FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINPASS The TSF shall detect when pinCount successful authentication events exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_PASS. FIA_AFL.1.2/ PINPASS When the defined number of successful authentication events has been met, the TSF shall block further authorization attempts. **FIA_AFL.1/PINFAIL Authentication failure handling** Hierarchic

SFR	Dependency	Rationale/ fulfilled by
		Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINFAIL The TSF shall detect when <u>pinCount</u> unsuccessful authentication attempts exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_FAIL. FIA_AFL.1.2/ PINFAIL When the

SFR	Dependency	Rationale/ fulfilled by
		defined number of unsuccessful authentication attempts has been met, the TSF shall block further authorization attempts. **FIA_UID.1**
FDP_ACC.1/M&R	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/M&R
FDP_ACF.1/M&R	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/M&R, FMT_MSA.3/M&R
FMT_MSA.1/M&R	[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	Fulfilled by FDP_ACF.1/M&R, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/M&R	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/M&R, FMT_SMR.1
FCO_NRO.1/M&R	FIA_UID.1 Timing of identification	Fulfilled by FIA_AFL.1**/PINPASS Authentication failure handling** Hierarchic

SFR	Dependency	Rationale/ fulfilled by
		Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINPASS The TSF shall detect when <u>pinCount</u> successful authentication events exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_PASS. FIA_AFL.1.2/ PINPASS When the defined number of successful authentication events

SFR	Dependency	Rationale/ fulfilled by
		has been met, the TSF shall block further authorization attempts. **FIA_AFL.1/PINFAIL Authentication failure handling** Hierarchic Dependencies: FIA_UAU.1 Timing of authentication. FIA_AFL.1.1/PINFAIL The TSF shall detect

SFR	Dependency	Rationale/ fulfilled by
		when pinCount unsuccessful authentication attempts exceeds pinLimit for an NV Index with the attribute TPM_NT_PIN_FAIL. FIA_AFL.1.2/ PINFAIL When the defined number of unsuccessful authentication attempts has been met, the TSF shall block further authorization attempts. **FIA_UID.1**
FDP_RIP.1	No dependencies	n. a.
FPT_FLS.1/FS	No dependencies	n. a.
FPT_FLS.1/SD	No dependencies	n. a.
FPT_PHP.3	No dependencies	n. a.
FDP_ITT.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/AC, FDP_ACC.1/Hier, FDP_ACC.1/NVM, FDP_ACC.1/ExIm, FMT_MSA.1/Cre, FDP_ACC.1/M&R, FDP_ACC.2/States
FPT_ITT.1	[FDP_ACC.1 Subset access control, or	Fulfilled by

SFR	Dependency	Rationale/ fulfilled by
	FDP_IFC.1 Subset information flow control]	FDP_ACC.1/AC, FDP_ACC.1/Hier, FDP_ACC.1/NVM, FDP_ACC.1/ExIm, FMT_MSA.1/Cre, FDP_ACC.1/M&R, FDP_ACC.2/States
FDP_SDI.1	No dependencies	n. a.
FDP_ACC.1/Hier	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/Hier
FDP_ACF.1/Hier	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/Hier, FMT_MSA.3/Hier
FMT_MSA.1/Hier	[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	Fulfilled by FDP_ACC.1/Hier, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/Hier	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/Hier, FMT_SMR.1
FMT_MSA.4/Hier	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/Hier,

Rationales for dependencies that are not fulfilled:

(1) The firmware update procedure as kind of user data import is realised based on commands that transfer single data packets into the TOE. No secure channel will be established and used for that process, the protection of the user data is done based on checks of each single packet. Hence the SFRs regarding trusted channel and trusted path are not applicable.

(2) The SFR FDP_ITC.2/ExIm addresses export and import of user data with security attributes. The data consistency is ensured because the other trusted IT product is always a system that is equivalent to the TOE. Especially the same TPM or another TPM may be used for import of exported data. Hence the SFR FPT_TDC.1 is not applicable.

(3) The exported and imported user data is based on data objects and not channel-based. The security attributes are part of the exported object and the object is integrity and confidentiality protected. Hence the SFRs regarding trusted channel and trusted path are not applicable.

7.3.3 Assurance Rationale

This protection profile requires the TOE to be evaluated on Evaluation Assurance Level 4 (EAL4) as defined in CC [3] and augmented with ALC_FLR.1 and AVA_VAN.4 listed in table 10.

EAL4 was selected because the objective of the TOE is to provide developers or users with a moderate to high level of independently assured security in conventional commodity TOEs and assumes that developers or users are prepared to incur additional security-specific engineering costs. EAL4 permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources.

The developer and manufacturer ensure that the TOE is designed and fabricated so that the TSF achieves the desired properties and it requires a combination of equipment, knowledge, skill, and time to be able to derive design information or affect the development and manufacturing process which could be used to compromise security through attack. This is addressed by the SAR of the class ALC especially by the component ALC_DVS.1.

Further the AVA_VAN.4 requires the developer and the manufacturer to provide necessary evaluation evidence that the TOE fulfills its security objectives and is resistant to attack with **Moderate** potential. The component AVA_VAN.4 will analyze and assess the resistance of the TOE to attacks with **Moderate** attack potential.

EAL4 is also augmented with ALC_FLR.1 to track and correct the reported and found security flaws in the product.

The component AVA_VAN.4 Methodical vulnerability analysis has the following dependencies:

ADV_ARC.1 Security architecture description
ADV_FSP.2 Security-enforcing functional specification
ADV_TDS.3 Basic modular design
ADV_IMP.1 Implementation representation of the TSF
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

All these components are contained in the EAL4 package. The component ALC_FLR.1 Basic flow remediation has no dependencies. Therefore all these dependencies are satisfied by EAL4.

8. Appendix

8.1 Random Number Generator (informative)

The internal RNG shall comply with the NIST Special Publication 800-90A [18]. Hence, its primary nature is that of a deterministic random bit generator (DRBG). According to [18] the entropy is taken from a seed given as input to the DRBG mechanism. The DRBG can be reseeded in order to add new entropy to the internal state.

In the TPM architecture specification [7], the RNG architecture is given in section 11.4.10. As shown in figure 4 of [7], the RNG contains (at least) one entropy source in order to seed or reseed the DRBG. The entropy should be collected in a state register of the RNG that is not visible to an outside process or other TPM capability. Using the command TPM2_StirRandom, additional data can be injected into the status registers, but the security of the DRBG itself does not rely on the secrecy of this information.

In order to meet the certification requirements of the intended market, the quality metric of the RNG depends on the quality of the entropy source and the seed period: If the seeding takes place only on initialisation time, the resulting RNG is a pure deterministic RNG. On the other hand, if the reseeding mechanism ensures that the entropy inserted into the RNG always exceeds the amount of entropy that is taken as output from the RNG, the RNG can be seen as physical RNG. Also, the reseeding could be implemented on a periodic base. In that case the amount of output data taken from the RNG may be bigger than the amount of entropy that was injected by reseeding. In that case the character of the RNG is hybrid. In summary, the character of the RNG can be determined by choosing the seed period: An infinite seed period creates a deterministic RNG while a very short seed period creates a physical RNG.
Regarding the quality of the entropy source, the NIST Special Publication 800-90B [19] can be taken into consideration. It also contains testing strategies to determine the entropy provided by the entropy source.

8.2 Acronyms

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

Acronym	Description
TPM	Prefix for an indication passed from the system interface of the TPM to a Protected Capability defined in the TPM2 Library specification
AuthData	Authentication Data or Authorisation Data, depending on the context
CA	Certificate Authority
CFB	Cipher Feedback mode
CRTM	Core Root of Trust for Measurement
CTR	Counter-mode encryption
DA	Dictionary Attack
DAA	Direct Autonomous Attestation
DRBG	Deterministic Random Bit Generator
EAL	evaluated assurance level

Acronym	Description
ECB	Electronic Codebook
ECC	Elliptic Curve Cryptography
ECDAA	ECC-based Direct Anonymous Attestation
ECDH	Elliptic Curve Diffie-Hellman
EK	Endorsement Key
EPS	Endorsement Primary Seed
FIPS	Federal Information Processing Standard
FUM	Field Upgrade mode
HMAC	Hash Message Authentication Code
HW	Hardware Interface
I/O	Input/Output
IV	Initialisation Vector
KDF	key derivation function
MMIO	Memory Mapped I/O
NIST	National Institute of Standards and Technology
NV	Non-volatile
NVM	Non-Volatile Memory
OAEP	Optimal Asymmetric Encryption Padding
PCR	platform configuration register(s)
PK	Primary Key
PP	Physical Presence, Protection Profile
PPO	Platform Primary Object
PPS	Platform Primary Seed
PRIVEK	Private Endorsement Key
PRNG	Pseudo Random Number Generator
PUBEK	Public Endorsement Key
RNG	Random Number Generator
RSA	Algorithm for public-key cryptography. The letters R, S, and A represent the initials of the first public describers of the algorithm Rivest, Shamir and Adleman.
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
SHA	Secure Hash Algorithm
SPS	Storage Primary Seed
SRK	Storage Root Key
TCB	trusted computing base
TCG	Trusted Computing Group
TOE	Target of Evaluation
TPM	Trusted Platform Module
TPM2_	Prefix for a command defined in the TPM2 Library specification

Acronym	Description
UTC	Universal Time Clock

8.3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 5, CCMB-2017-04-001, April 2017

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 3.1, Revision 5, CCMB-2017-04-002, April 2017

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 5, CCMB-2017-04-003, April 2017

[4] Common Methodology for Information Technology Security Evaluation Methodology, Evaluation Methodology, Version 3.1, Revision 5, CCMB-2017-04-004, April 2017

[5] Common Criteria Recognition Arrangement Management Committee, Policies and Procedures, Supporting Documents for Smartcards and similar devices, document number 2006-06-001

[6] Supporting Document Guidance Smartcard Evaluation, February 2010, Version 2.0, CCDB-2010-03-001

[7] TPM Library Part 1: Architecture, Specification Version 2.0, Revision 1.38, September 2016, Trusted Computing Group, Incorporated

[8] TPM Library Part 2: TPM Structures, Specification Version 2.0, Revision 1.38, September 2016, Trusted Computing Group, Incorporated

[9] TPM Library Part 3: Commands, Specification Version 2.0, Revision 1.38, September 2016, Trusted Computing Group, Incorporated

[10] TPM Library Part 4: Supporting Routines, Specification Version 2.0, Revision 1.38, September 2016, Trusted Computing Group, Incorporated

[11] TCG PC Client Specific Platform TPM Profile for TPM 2.0 (PTP), Family “2.0”, Level 00 Revision01.03 August 2017,

[12] ITU-T X.690: Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

[13] FIPS-140-2, Federal Information Processing Standard 140-2

[14] FIPS-180-4, Federal Information Processing Standard 180-4 Secure Hash Standard (SHS)

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

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

[17] FIPS-197, Federal Information Processing Standard 197

[18] NIST Special Publication 800-90A: Recommendation for Random Number Generation Using Deterministic Random Bit Generators. January 2012

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

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

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

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

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

[24] IETF RFC 2104, Internet Engineering Task Force Request for Comments 2104: HMAC: Keyed-Hashing for Message Authentication

[25] IETF RFC 2119, Internet Engineering Task Force Request for Comments 2119: Key words for use in RFCs to Indicate Requirement Levels

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

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

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

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

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

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

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