

Trusted Computing Group
Protection Profile
PC Client Specific Trusted Platform Module
TPM Family 1.2; Level 2

Version: 1.1; July 10, 2008

This page is intentionally left blank.

Table of content

1.	PP Introduction	5
1.1.	PP Reference	5
1.2.	PP organization	5
1.3.	TOE Overview	5
1.3.1.	TOE Definition.....	5
1.3.2.	Security services.....	6
1.3.3.	TPM life cycle.....	8
1.3.4.	User, Subjects, Objects and Operations.....	10
2.	Conformance Claims	22
2.1.	CC Conformance Claim.....	22
2.2.	PP Claim.....	22
2.3.	Package Claim	22
3.	Extended Components Definition	23
3.1.	Family Random Number Generation.....	23
4.	Security Problem Definition	24
4.1.	Threats	24
4.2.	Organisational Security Policies	25
4.3.	Assumptions	26
5.	Security Objectives	27
5.1.	Security Objectives for the TOE	27
5.2.	Security Objectives for the Operational Environment.....	29
5.3.	Security Objectives Rationale.....	30
6.	Security Requirements	40
6.1.	Security Functional Requirements for the TOE	40
6.1.1.	General SFR.....	40
6.1.2.	Cryptographic support.....	42

6.1.3.	TPM Operational Modes	46
6.1.4.	Identification, Authentication and Binding	52
6.1.5.	Data Protection and Privacy	60
6.1.6.	Data Import and Export.....	82
6.1.7.	DAA.....	91
6.1.8.	TSF Protection	93
6.2.	Security Assurance Requirements for the TOE.....	96
6.3.	Security Requirements Rationale	97
6.3.1.	Rationale for the Security Functional Requirements.....	99
6.3.2.	Rationale for the Security Assurance Requirements	109
6.3.3.	SFR Dependency Rationale	110
7.	Annex.....	118
7.1.	Ordinal table	118
7.2.	Acronyms.....	123
7.3.	Glossary	124
7.4.	Literature	128
8.	Optional Package “Revoke of Trust” (Informative Annex).....	130

1. PP Introduction

1.1. PP Reference

Title: Protection profile PC Client Specific Trusted Platform Module Family 1.2; Level 2 (PP TPM F1.2L2)

Version: 1.1; July 10, 2008

Author: Trusted Computing Group

Publication date: TBD

1.2. PP organization

This protection profile (PP) describes security requirements for the Trusted Computing Group (TCG) PC Client Specific Trusted Platform Module (TPM) Family 1.2; Level 2 conforming to the Common Criteria version 3.1 release 2. The TOE of the current PP is a PC Client Specific TPM conforming to the TPM specification version 1.2, level 2 [5] [6] [7], where the TOE of the TPM PP [10] is a TPM conforming to the TPM specification version 1.1b. The update of the TPM specification introduces new security features, deprecated and deleted commands which affect the current TPM PP in the security problem definition, the security objectives and the security requirements.

The informal annex of this PP provides an optional package for the revocation of trust as optional security feature of the TPM.

The current TPM PP is conforming to Common Criteria version 3.1 whereas the TPM PP [10] conforms to the Common Criteria version 2.1. This results in editorial changes of the common security requirements which are common to both PP.

1.3. TOE Overview

1.3.1. TOE Definition

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

The primitives provided by the TOE include cryptographic algorithms for key generation, digital signatures, random number generation, sealing data to system state, protected storage, binding information to the TPM, support of direct anonymous attestation and physical protection. Attestation by the TOE is an operation that provides proof of data known

to the TPM. This is done by digitally signing specific internal TPM data using an Attestation Identity Key (AIK). The acceptance and validity of both the integrity measurements and the AIK itself are determined by the Verifier. The AIK is obtained using either the Privacy Certification Authority or the Direct Anonymous Attestation (DAA) protocol. The DAA is a protocol for vouching for an AIK using zero-knowledge-proof technology.

1.3.2. Security services

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

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

Support for the Root of Trust for Measurement

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

Root of Trust for Reporting

The root of trust for reporting (RTR) exposes the measurement digests stored in the PCRs and attests to the authenticity of these measurement digests based on trusted platform identities or the Direct Anonymous Attestation Protocol. The trusted platform identities for RTR are defined by Attestation Identity Credentials for Attestation Identity Keys (AIK) ¹ generated by the TPM. The TPM creates digital signatures over the PCR values using an Attestation Identity Key.

¹ Cf. glossary section 7.3 for details

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

- An Endorsement Key is bound to one and only one TPM (i.e., there is a one to one correspondence between an Endorsement Key and a TPM.)
- A TPM is bound to one and only one Platform, (i.e., there is a one to one correspondence between a TPM and a Platform.)
- Therefore, an Endorsement Key is bound to a Platform, (i.e., there is a one to one correspondence between an Endorsement Key and a Platform.)

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

Root of Trust for Storage

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

The functionality used to provide secure storage is:

- TPM_Seal and TPM_Unseal, which perform RSA encrypt and decrypt, respectively, on data that is externally generated. The sealing operation encrypts not only the data, but also the values of the selected PCRs and the locality that must exist during for unseal and tpmProof, which is a unique secret identifier for the TPM sealing the data. To unseal the data, three conditions must exist: 1) the appropriate key must be available for unseal, 2) the TPM PCRs must contain the values defined at the time of the seal operation, and 3) the value of tpmProof must be the same as that encrypted during the seal operation. By requiring the PCR values to be duplicated at unseal and the tpmProof value to be checked, the seal operation allows software to explicitly state the future "trusted" configuration that the platform must be in for the decrypted key to be used and for decryption to only occur on the specified TPM.
- TPM_Unbind, which RSA decrypts a blob created outside the TPM that has been encrypted using a public key where the associated private key is stored in the TPM.

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

- The Storage Root Key (SRK), which is the root key of a hierarchy of keys associated with a TPM; it is generated within a TPM and is a non-migratable key. Each owned TPM contains a SRK, generated by the TPM at the request of the Owner. Under that SRK may be organized different trees dealing with migratable data or non-migratable data.

- Storage keys, which are used to RSA encrypt and RSA decrypt other keys and sealed data with their security attributes in the Protected Storage hierarchy, only.
- Binding Keys, which are used for TPM_Unbind operations only. A binding operation (performed outside the TPM) associates identification and authentication data with a particular data set and the entire data blob is encrypted outside the TPM using a binding key, which is an RSA key. The TPM_Unbind operation uses a private key stored in the TPM to decrypt the blob so that the data (often a key pair) stored in the blob may be used.

Other security services and features

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

The TPM provides non-volatile storage as a shielded location for data of external entities. The TPM owner controls access to the non-volatile storage. The access control may include the need for authentication of the user, delegations, PCR values and other controls.

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

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

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

1.3.3. TPM life cycle

The TPM life cycle has 7 phases from the protection profile prospective:

1. TPM development
2. TPM manufacturing
3. Platform manufacturing and delivery
4. Platform deployment phase

5. Platform identity registration
6. Platform operation
7. Platform recycling and retirement

The Figure 1 shows typical activities of the TPM life cycle Phase 1 to Phase 7. The Phase 1 and 2 are TOE development and manufacturing. They are subject of the evaluation of the development environment. The functions in the white area are implemented (e.g. or at least supported) by the TOE (e.g. EK may be generated by the TOE or injected into TOE). The grey area shows activities in the TOE environment.

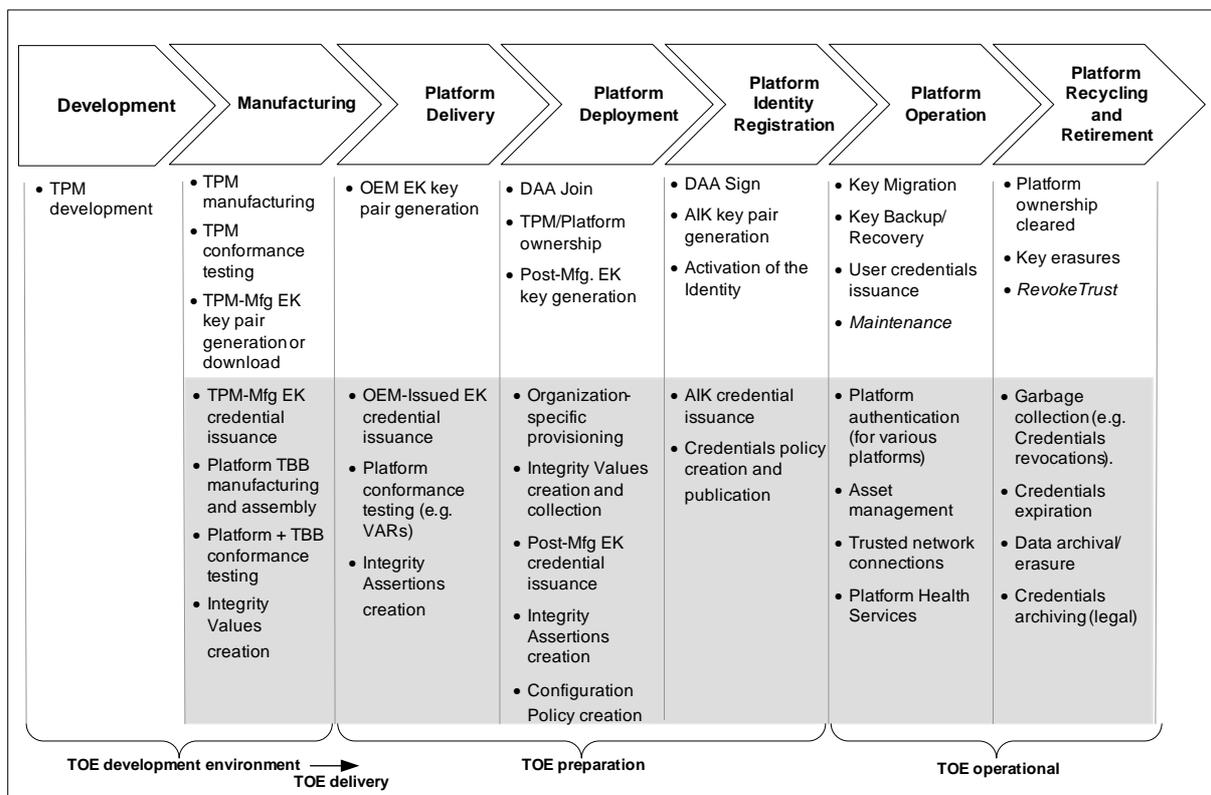


Figure 1: TOE lifecycle

The whole life-cycle of the TPM will be considered during evaluations based on this Protection Profile as far as the developer/manufacturer of the TOE is directly involved.

The scope of the assurance components referring to the product's life-cycle is limited to Phases 1 and 2. These phases are under the control of the TPM developer and the TPM manufacturer. This includes the interfaces to the other phases where information and material is being exchanged with the partners of the developer/manufacturer of the TOE. The TPM manufacturer may use the TOE security functions like endorsement key generation described by security functional requirements.

The security functional requirements addressed in this protection profile are mainly used in the Phase 3 to Phase 7.

Application note 1: The intention of the PP is to consider phases 1 and 2 as part of the evaluation and therefore define TOE delivery according to CC as after phase 2 or later. The initialization process and its environment may depend on specific security needs of a platform developer or costumer. The security target shall describe the instantiation of the life cycle defined in this PP that is relevant for the product evaluation process. It is important to define the point of TOE delivery in the life cycle required for the evaluation according to CC requirements ALC_DEL. All development and manufacturing steps before TOE delivery have to be part of the evaluation under assurance class ALC. This includes generating and loading the endorsement key into the TPM if this manufacturing option is used for the TOE. All production, generation and installation procedures between TOE delivery and operational usage have to be considered in the product evaluation process under AGD_PRE assurance family. Therefore, the Security Target has to outline the partition the TPM development environment and the TOE operational environment and the related security objectives for the TOE development into aspects that are relevant before and after TOE delivery.

1.3.4. User, Subjects, Objects and Operations

User roles

A user is any active entity outside of the TOE. Six operational roles of users are defined for the TOE.

(1) TPM owner

The TPM owner controls the use of the Endorsement Key and of the Storage Root Key. The TPM owner creates and controls the Attestation Identity Keys. The TPM Owner does the selection and the authorization of migration authorities at any time prior to key migration. The TPM owner may partly delegate his authorization to other users.

(2) Delegated entity

The TPM owner or another delegated entity (within their authorization) may delegate its authorization for selected commands to a delegated entity by means of a delegation table in a delegation blob. The delegation table lists the command ordinals allowable for use by the delegate, the identity of a process that can use the ordinal list, and the AuthData value to use the ordinal list (cf. to [5] chapter 29 for details).

(3) Entity owner

The user creating an entity and defining the entity owner token AuthData as security attribute of this entity by means of the AuthData Insertion Protocol (ADIP). The presentation of the entity owner token is required for loading this entity into the TPM.

(4) Entity user

Entity user uses a loaded entity. The authentication of more than one entity user, i.e., verification of knowledge of the entity user token (i.e. usageAuth), may be required as a condition of using a loaded entity.

(5) User using operatorAuth

User under physical presence may define authorization data operatorAuth for a user role allowed to deactivate temporarily the TPM.

(6) World

The role "World" is assigned to any user not authenticated for any role listed above.

The TPM knows some user roles by permanently stored individual authentication data: the TPM owner using ownerAuth and a user role defined by operatorAuth2. Other user roles are object specific (entity owner and entity user). The users may use these roles except "World" after authentication only. The user establishing an OSAP session negotiates the authorization handle and the shared secret as user authentication data during this session.

The user have the following security attributes which are indicated to the TPM in a platform specific way

- Physical presence

The TCG policies require physical presence to be indicated to the TPM to enable the user to execute privileged commands³ or to alter the following states if no TPM owner is defined: disabled, deactivated, and (ownership-)disabled (cf. [5], chapter 10, and section 6.1.3 TPM Operational Modes of this PP for further details). The TPM and platform manufacturer define the way how the TPM physical presence is asserted by hardware input. The physical presence may be asserted by means of the command TSC_PhysicalPresence (cf. [7], section 6.6).

- Locality

When a platform is designed with a trusted process, the trusted process may wish to communicate with the TPM and indicate that the command is coming from the trusted process. The commands that the trusted process sends to the TPM are the normal TPM commands with a Locality modifier that indicates that the trusted process initiated the command. The TPM accepts the command as coming from the trusted process merely due to the fact that the modifier is set (cf. [5], chapter 16, for further details).

² This user is some times referred to as "operator".

³ cf. the ordinal table 12, column "Physical presence"

Subjects

In the context of this PP the subjects are:

- the initialization process initiated by the TPM_Init signal of the platform to the TPM,
- the self test which is started after initialization and continued after the TPM_ContinueSelfTest command is received,
- a process receiving, executing and responding to a single command,
- a session (sequence of commands) which may establish shared secrets, encryption keys, and session logs.

The session may be an authorization session (cf. [5], ch. 13), monotonic counter sessions (cf. [5], sec. 20.1), a tick session (cf. [5], sec. 20.1) or a transport session (cf. [5], chapter 18).

In order to communicate with a subject, users shall first associate themselves with that subject, through a process called binding. Binding is established with

- the user "World" by default,
- other users by means of authorization protocols as described in the TPM Main Part 1 Design Principles [5], chapter 13.

The subject is associated with the security attributes

- the authData, locality and physical presence presented by the user,
- the rolling nonce, authorization handles if the subject is a session,
- the authorization handle and the shared secret if the subject is a OSAP session,
- the authorization associated with the delegation blob if the subject is a DSAP session,
- the current PCR values.

The Object Specific Authorization Protocol (OSAP) session is bound to an object. The Object-Independent Authorization Protocol (OIAP) session is independent on any object.

Objects and Operations

The data in the shielded location and the protected capabilities of the TPM are the TOE assets. The objects treated by the TOE are the data generated or stored in the shielded location or to be imported into or to be exported from the shielded location. The operations of the TOE are the protected capabilities of the TPM. They are defined by the TPM commands (cf. [7]).

The TPM holds operational mode flags as security attributes for all objects stored in the TPM_PERMANENT_FLAGS (cf. [6] sec. 7.1) and TPM_STCLEAR_FLAGS (cf. [6] sec. 7.2)⁴:

- disable: flag whether the TPM is enabled (FALSE) / disabled (TRUE),
- deactivated: flag whether the TPM is active (FALSE) / inactive (TRUE),
- owner: flag indicating whether an owner is installed, unowned (TRUE) / owned (FALSE)⁵.

The default value of “disable”, “deactivated” and “unowned” is TRUE.

The security attributes associated with data objects are stored as TPM permanent data, flags in the TPM and other TPM resources (e.g. NV storage) or associated with data in an encrypted blob. Note that the authorization data associated with an object define the authorized user and the related authentication reference data for this user.

The table 1 list the objects, the operation via reference to the commands as described in the TPM specification [7] and the security attributes of the objects as described in the TPM specification [6].

Table 1: Objects, operations, security attributes and authorization data

#	Object	Operation	Security attributes and authorization data
1	EK ⁶ The non-revocable TPM Endorsement	generate : generate an permanent EK (cf. cmd TPM_CreateEndorsementKeyPair [7], sec. 14.1) ⁷ .	ownerAuth : authorization data to read public key part and to use EK, defined in TPM_PERMANENT_DATA

⁴ This PP uses the names of the security attributes e.g. “disabled” and refers to the structure where they are stored (i.e. TPM_PERMANENT_FLAGS for the security attribute “disabled”). The structures are described in the TPM specification part 2 [6]. The security attributes are used in the commands described in the TPM specification part 3 [7].

⁵ The TPM specification does not define a specific flag indication whether a TPM owner is installed or not in [6] sec. 7.1. The TPM_TakeOwnership inserts the TPM Ownership value into the TPM. The TPM_OwnerClear and TPM_ForceClear clears (“unowns”) the TPM owner value. The TPM_TakeOwnership decides on existence of a valid ownerAuth whether the TPM is owned or not. The readPubek flag is set to FALSE by TPM_TakeOwnership and set to TRUE by TPM_OwnerClear and TPM_ForceClear, thus mirroring if a TPM Owner is present (cf. [7], sec. 14.4).

⁶ This PP addresses permanent (non-revocable) EK only. Revocation of trust and a revocable EK are optional features described in the package Revoke of Trust, cf. informative annex of this PP.

⁷ Note the TPM specification [5] and [7] allows for the manufacturing option to generate the EK outside the TPM or to “squirt” the EK into the TOE. Because the PP defines mandatory SFR only it was decided that the PP does not include security functional requirements for generation or “squirted” the EK. The ST shall state whether generation or “squirted” of the EK (cf. [5], ch. 5) is

#	Object	Operation	Security attributes and authorization data
	<p>Key (EK) is an asymmetric RSA key pair which is uniquely associated with each TPM device over the TPM life time.</p>	<p>14.1)⁷.</p> <p>use: take ownership (cf. cmd TPM_TakeOwnership, [7], sec. 6.1), decryption of AIK credentials (cf. cmd TPM_ActivateIdentity, [7], sec. 15.2), prove a DAA attestation held by a TPM (cf. cmd TPM_DAA_Sign, [7], sec. 26.2)</p> <p>read: exports the public portion of the EK (cf. cmd TPM_ReadPubek [7], sec. 14.4, TPM_OwnerReadInternalPub [7], sec. 14.5)</p>	<p>(cf. [6], sec. 7.4)</p> <p>readPubek: ability to read the PUBEK without owner AuthData (the default state is TRUE), defined in TPM_PERMANENT_FLAGS (cf. [6], sec. 7.1)</p> <p>CEKPUSED: indication whether the EK is “squirted” or generated, defined in TPM_PERMANENT_FLAGS (cf. [6], sec. 7.1)</p>
2	<p>SRK</p> <p>The Storage Root Key (SRK) is the RSA highest key pair of the key hierarchy for encryption respective decryption of keys.</p>	<p>generate: generate the SRK (cf. cmd TPM_TakeOwnership [7], sec. 6.1)</p> <p>use: wrap (encrypt) and unwrap (decrypt) other keys in the Protected Storage hierarchy (cf. [7], ch. 10)</p> <p>delete: Delete the SRK (cf. cmd TPM_OwnerClear, [7], sec. 6.2; TPM_ForceClear [7], sec. 6.3)</p> <p>read: export of the public part (cf. cmd TPM_GetPubKey [7], sec. 10.6, TPM_OwnerReadInternalPub [7], sec. 14.5)</p>	<p>ownerAuth: authorization data to read public key part or to delete the SRK (cf. [6], sec. 7.4 to read public key part)</p> <p>srkAuth: authorization to use the SRK for the cmd TPM_MakeIdentity (cf. [7], sec. 15.1)</p> <p>srkParams: key parameter set by cmd TPM_TakeOwnership (cf. [7], sec. 6.1)</p> <p>ReadSRKpub: if TRUE: reading the SRK pub key is allowed, if FALSE: reading the SRK pub key is not allowed, Default is FALSE. (cf. [6], sec. 7.1)</p> <p>See also the flags ownership, disableOwnerClear and</p>

used for the TOE and the evaluation will assess the security of this process in the evaluation sub-activities ALC_DVS.1 Identification of security measures and ALC_DEL.1 Delivery procedures.

#	Object	Operation	Security attributes and authorization data
			disableForceClear defined in TPM_PERMANENT_FLAGS (cf. [6], sec. 7.1)
3	<p>User Key</p> <p>Any cryptographic key except the EK and the SRK.</p>	<p>create: the subject generates a key and exports a wrapped key blob of this key (cf. cmd TPM_MakeIdentity [7], sec. 15.1, TPM_CreateWrapKey [7], sec. 11.4, TPM_CMK_CreateKey [7], sec. 11.7),</p> <p>use: the subject performs a cryptographic operation depending on the command encryption, decryption or signing (cf. [6], sec. 5.8),</p> <p>import: the subject loads a key in form of a wrapped key blob (cf. command TPM_ActivateIdentity [7], sec. 15.2, TPM_LoadKey2, [7], sec. 10.5, TPM_LoadKey, [7], sec. 27.9),</p> <p>export: the subject creates and exports the newly created key in form of a wrapped key blob or exports the public key of a loaded key (cf. cmd TPM_MakeIdentity [7], sec. 15.1, TPM_CreateWrapKey [7], sec. 10.4, or TPM_CMK_CreateKey [7], sec. 11.7, TPM_GetPubKey [7], sec. 10.6)</p> <p>delete: delete the key in the internal storage of the TPM (cf. cmd TPM_FlushSpecific [7], sec. 22.1, TPM_EvictKey [7], sec. 27.1.1)</p>	<p>authDataUsage: indication whether authorization sessions for an key are required, defined in TPM_AUTH_DATA_USAGE (cf. [6], sec. 10.2),</p> <p>usageAuth: authorization data for use, defined in TPM_STORE_ASYMKEY (cf. [6], sec. 10.6)</p> <p>migrationAuth: authorization data for migration, defined in TPM_STORE_ASYMKEY (cf. [6], sec. 10.6),</p> <p>keyUsage: The selection of a key usage value limits the choices of encryption and signature schemes, defined in TPM_KEY_USAGE (cf. [6], sec. 5.8),</p> <p>algorithmParms: identifies the cryptographic algorithm and its parameter, defined in TPM_KEY_PARMS (cf. [6], sec. 10.1),</p> <p>keyFlags: describes the operations allowed for the key, defined in TPM_KEY_FLAGS (cf. [6], sec. 5.10),</p> <p>PCRInfo (optional): value of selected PCR to control import and access to keys, defined in TPM_PCR_INFO in case of TPM_KEY (cf. [6], sec. 5.2), and TPM_PCR_INFO_LONG in</p>

#	Object	Operation	Security attributes and authorization data
			<p>case of TPM_KEY12 (cf. [6], sec. 5.3)</p> <p>OwnerEvict: keys under control of the OwnerEvict flag must stay resident in the TPM (cf. to TPM_KEY_CONTROL_OWNER_EVICT bit in [6], sec.10.9, and the cmd TPM_KeyControlOwner, [7], sec. 21.1, TPM_FlushSpecific, [7], sec. 22.1, and TPM_EvictKey [7], sec. 27.1)</p>
4	<p>Monotonic counter</p> <p>The monotonic counter provides an ever-increasing incremental value to the user.</p>	<p>create: create a monotonic counter (cf. cmd TPM_CreateCounter, [7], sec. 25.1),</p> <p>read: read the value of the monotonic counter (cf. cmd TPM_ReadCounter, [7], sec. 25.1),</p> <p>increment: increment the value of the monotonic counter (cf. cmd TPM_IncrementCounter, [7], sec. 25.2),</p> <p>release: delete the monotonic counter by cmd TPM_ReleaseCounter (cf., [7], sec. 25.4) and TPM_ReleaseCounterOwner (cf. [7], sec. 25.5)</p>	<p>countId: identity of the counter for reference in the cmd using the counter (cf. [6], sec. 7.5),</p> <p>authHandle: authorization session handle used for owner authentication. (cf. [6], sec. 5.6)</p>
5	<p>NV storage</p> <p>Non-volatile storage of the TPM provided to the user and protected by access rights managed by the TPM owner.</p>	<p>create: create the NV storage area and define its security attributes (cf. cmd TPM_NV_DefineSpace [7], sec. 25.1),</p> <p>read: read the value of a register in the NV storage area (cf. cmd TPM_NV_ReadValue [7], sec. 25.1, TPM_NV_ReadValueAuth [7], sec. 25.2.),</p>	<p>authValue: authorization session handle used for NV storage (cf. [6], sec. 19.4),</p> <p>noOwnerNVWrite: The count of NV writes that have occurred when there is no TPM Owner. This value starts at 0 in manufacturing and after each TPM_OwnerClear. If the value exceeds 64 the TPM returns TPM_MAXNV-</p>

#	Object	Operation	Security attributes and authorization data
		<p>write: write a value into a register in the NV storage area (cf. cmd TPM_NV_WriteValue [7], sec. 25.2, TPM_NV_WriteValueAuth [7], sec. 25.3),</p>	<p>WRITES to any command attempting to manipulate the NV storage. (cf. TPM_PERMANENT_DATA in [6], sec. 7.4)</p> <p>nvLocked: flag indicating the activation of authorization checks. TRUE: All NV area authorization checks are active FALSE: No NV area checks are performed, except for maxNVWrites. FALSE is the default value (cf. TPM_PERMANENT_FLAGS in [6], sec. 7.1).</p> <p>pcrInfoRead and pcrInfoWrite: values of selected PCR to control access to the NV storage, defined in TPM_NV_DATA_PUBLIC (cf. [6], sec. 19.3) including localityAtRelease, which controls read and write access depending on locality (cf. TPM_PCR_INFO_SHORT in [6], sec. 8.5)</p> <p>permissions: permissions to manipulate the area, defined in TPM_NV_ATTRIBUTES (cf. [6], sec. 19.2)</p> <p>bReadSTClear, bWriteSTClear; bWriteDefine: flags defined in TPM_NV_DATA_PUBLIC (cf. [6], sec. 19.3)</p>
6	<p>PCR Platform Configuration</p>	<p>reset: reset the PCR value to a well defined value (cf. cmd TPM_Startup [7], sec. 3.2,</p>	<p>pcrReset, pcrResetLocal, pcrExtendLocal: flags as defined in TPM_PCR_ATTRI-</p>

#	Object	Operation	Security attributes and authorization data
	Register (PCR) intended to record measurement digests and to be used for attestation and access control.	<p>TPM_PCR_Reset [7], sec. 16.1),</p> <p>read: read the value of a PCR (cf. cmd TPM_PCRRead [7], sec. 16.2),</p> <p>quote: sign with an identified signing key and export the values of identified set of PCR (cf. cmd TPM_Quote [7], sec. 13.3, TPM_Quote2 [7], sec. 15)</p> <p>extend: calculate the hash value of the PCR value and the result of a pending hash calculation (cf. cmd TPM_SHA1CompleteExtend [7], sec. 13.4) or an presented data (cf. cmd TPM_Extend, [7], sec. 13.1) and the LPC commands TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END (cf. [8], ch. 8).⁸</p>	BUTES (cf. [6], sec. 8.8)
7	RNG The TPM random number generator (RNG) creates random numbers provided to the user and for internal use (e.g. key generation, secrets, nonce).	<p>read: read the next random number generated by the TPM (cf. cmd TPM_GetRandom [7], sec. 7.1),</p> <p>refresh: provides any data as input to the random number generator to refresh the internal state of the random number generator (cf. cmd TPM_StirRandom [7], sec. 7.2)</p>	No security attributes
8	Migration Key Blob Data object containing a User Key to be migrated to another TPM.	create: create a Migration Key Blob for an imported warped key blob (cf. cmd TPM_CreateMigrationBlob, [7], sec. 11.1),	<p>payload type = TPM_PT_MIGRATE</p> <p>migrationAuth: migration authorization data, defined in TPM_STORE_ASYMKEY (cf.</p>

⁸ Note the PCR are used internally to control access by the commands TPM_Seal, TPM_Sealx, ... as well.

#	Object	Operation	Security attributes and authorization data
	TPM.	migrate: (cf. cmd TPM_MigrateKey, [7], sec. 11.4), convert: (cf. cmd TPM_ConvertMigrationBlob, [7], sec. 11.2)	[6], sec. 10.6) authDataUsage, KeyUsage, algorithmParms, keyFlags, PCRInfo (optional): see User Key (cf. TPM_KEY, [6], sec. 10.2)
9	Certified Migration Key Blob Data object containing a User Key to be migrated to another TPM under control of a migration authority can be controlled (cf. Certified Migration Key Type [7] ch. 37).	create: (cf. cmd TPM_CMK_CreateBlob [7], sec.11.9), migrate: (cf. cmd TPM_MigrateKey, [7], sec. 11.4), convert: (cf. cmd TPM_CMK_ConvertMigration, [7], sec. 11.10)	payload type = TPM_PT_CMK_MIGRATE authDataUsage, KeyUsage, algorithmParms, keyFlags, PCRInfo: see User Key as defined in TPM_KEY12 (cf. [6], sec. 10.33) migrationAuth: migration authorization data, defined in TPM_STORE_ASYMKEY (cf. [6], sec. 10.6, and TPM_CMK_AUTH in [6], sec. 5.16 for details)
10	Context A TPM resource cached outside the TPM. The resource may be a User Key, a authorization session context, a transport session context or a DAA TPM specific blob.	save (cf. cmd TPM_SaveContext [7], sec. 21.2, TPM_SaveKeyContext [7], sec. 27.12.1, TPM_SaveAuthContext [7], sec. 27.12.3), load (cf. cmd TPM_LoadContext [7], sec. 21.3, TPM_LoadKeyContext [7], sec. 27.12.2, TPM_LoadAuthContext [7], sec. 27.12.4)	resourceType : type of data stored in the context (cf. TPM_STANY_DATA, [6], sec. 7.6) tpmProof: the context is bind to the TPM by including the unique tpmProof (cf. TPM_PERMANENT_FLAGS in [6], sec. 7.1) in the integrityDigest (cf. e.g. cmd TPM_LoadContext [7], sec. 21.3)
11	Tick Counter The number of timer ticks the TPM has counted from the start of a timing session.	read: reading the counter value (cf. cmd TPM_GetTicks [7], sec. 5.2), create time stamp (cf. cmd TPM_TickStampBlob [7], sec. 5.2)	Tick Session Nonce (TSN): unique name of the tick counter defined in TPM_CURRENT_TICKS (cf. [6], sec. 15.1). Tick Increment Rate (TIR): rate at which the TCV is incremented per second

#	Object	Operation	Security attributes and authorization data
			defined in TPM_CURRENT_TICKS (cf. [6], sec. 15.1).
12	<p>Sealed Data</p> <p>An encrypted data blob which the TPM will import and decrypt if a selected set of PCR and tpmProof match specified values and the key for decryption is available.</p>	<p>export: (cf. cmd TPM_Seal, [7], sec. 10.1, and TPM_Sealx, [7], sec. 10.7),</p> <p>import: (cf. cmd TPM_Unseal, [7], sec. 10.2),</p>	<p>payload type = TPM_PT_SEAL</p> <p>AuthData: authorization data for import of the sealed data (cf. [6], sec. 9.3),</p> <p>tpmProof: the context is bind to the TPM by including the unique tpmProof (cf. TPM_MANENT_FLAGS in [6], sec. 7.1) in the integrity-Digest (cf. [6], sec. 9.3),</p> <p>PCRInfo (optional): value of selected PCR to control import and access to the sealed data, defined as TPM_PCR_INFO (cf. [6], sec. 8.3) (or none PCR information) in case of TPM_STORED_DATA (cf. [6], sec. 9.1) or TPM_PCR_INFO_LONG (cf. [6], sec. 8.4) in case of TPM_STORED_DATA12 (cf. [6], sec. 9.2)</p>
13	<p>Bound Blob</p> <p>A data blob encrypted under an asymmetric key</p>	<p>unbind: import of an encrypted data blob (cf. cmd TPM_Unbind [7], sec. 10.3)</p>	<p>payload type = TPM_PT_BIND</p> <p>none security attributes</p>

TSF data and user data

The objects listed in table 1 are user data except the EK and SRK.

The TSF data are

- EK and SRK,

- authentication reference data of the TPM owner, the entity using operatorAuth, delegated entities, owner of entities, user of entities, and
- TPM flags disable, deactivated and ownership as part of the TPM_PERMANENT_FLAGS,
- security attributes of the objects and subjects.

2. Conformance Claims

Conformance statement: this PP requires **strict** conformance of any ST or PP, which claims conformance to this PP.

2.1. CC Conformance Claim

This Protection Profile claims to be conformant with the Common Criteria version 3.1 Release 2.

This PP claims to be conformant to Common Criteria Part 2 [2] extended and to Common Criteria part 3 [3].

2.2. PP Claim

This PP does not claim conformance to any other PP.

2.3. Package Claim

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

3. Extended Components Definition

3.1. Family Random Number Generation

To define the IT security functional requirements of the TOE an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined here. This family describes the 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 behaviour

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

Component levelling:



FCS_RNG.1	Generation of random numbers requires that random numbers meet a defined quality metric.
Management:	FCS_RNG.1 There are no management activities foreseen.
Audit:	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: <i>physical, non-physical true, deterministic, hybrid</i>] random number generator that implements: [assignment: <i>list of security capabilities</i>].
FCS_RNG.1.2	The TSF shall provide random numbers that meet [assignment: <i>a defined quality metric</i>].

4. Security Problem Definition

4.1. 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 2: 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 authorized to perform.
2	T.Bypass	An unauthorized individual or user may tamper with TSF, security attributes or other data in order to bypass TOE security functions and gain unauthorized access to TOE assets.
3	T.Export	An 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 unauthorized individual or user to compromise keys generated within the TPM or encrypted data or to modify data undetected.
5	T.Hack_Physical	An unauthorized individual or user of the TOE may cause unauthorized disclosure or modification of TOE assets by physically interacting with the TOE. The attacker may be a hostile user of the TOE.
6	T.Imperson	An unauthorized individual may impersonate an authorized user of the TOE (e.g. by dictionary attacks to guess the authorization data) and thereby gain access to TOE data in shielded location and protected capabilities.
7	T.Import	A user or attacker may import data without security attributes or with erroneous security attributes, causing key ownership and authorization to be uncertain or erroneous 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.

#	Threat	Description
9	T.Intercept	An attacker may intercept the communication between a user and the TPM subjects to gain knowledge of the commands and data sent to the subject or manipulate the communication.
10	T.Malfunction	TOE assets may be modified or disclosed to an unauthorized individual or user of the TOE, through malfunction of the TOE.
11	T.Modify	An attacker may modify data in shielded locations or their security attributes 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 unauthorized changes to security attribute values for an object to enable attacks.
13	T.Replay	An unauthorized individual may gain access to the system and sensitive data through a "replay" or "man-in-the-middle" attack that allows the individual to capture identification and authentication data.
14	T.Repudiate_Transact	An originator of data may deny originating the data to avoid accountability.
15	T.Residual_Info	A user may obtain information that the user is not authorized to have when the data in shielded locations is no longer actively managed by the TOE ("data scavenging").

4.2. Organisational Security Policies

This section of the security problem definition shows the Organizational 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 organization 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 3: Organizational Security Policies

#	OSP	Description
1	OSP.Anonymity	Authorized users shall be able to hide temporarily the TPM attestation identity.
2	OSP.Context_Management	A resource manager shall be able to secure caching of resources without knowledge or assistance from the application that loaded the resource.
3	OSP.Delegation	The TPM supports multiple trusted processes obeying the principle of least privilege by means of role based administration and separation of duty by allowing delegation of individual TPM

#	OSP	Description
		Owner privileges to individual entities, which may be trusted processes.
4	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.
5	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.
6	OSP.RT_Reporting	The root of trust for reporting attests the authenticity of measurement digests based on trusted platform identities by means of digital signatures with the certified AIK.
7	OSP.RT_Storage	The root of trust for storage protects the keys and data entrusted to the TPM in confidentiality and integrity.
8	OSP.Anonymous_Attestation	The DAA issuer and the TPM owner establish a procedure for attestation without revealing the attestation information (i.e. the identity of the TPM).

4.3. Assumptions

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

Table 4: Assumptions to the IT environment

#	Assumption	Description
1	A.Configuration	The TOE will be properly installed and configured.
2	A.Physical_Presence	The Host Platform's trusted processes assert physical presence of local operator to the TPM.

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 5: Security objectives for the TOE

#	Objective	Description
1	O.Anonymity	The TOE must allow the user authenticated by operatorAuth and the user "World" under physical presence temporarily to deactivate the TPM and to hide the TPM attestation identity during a user session.
2	O.Context_Management	The TOE must ensure a secure wrapping of a resource (except EK and SRK) 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.
3	O.Crypto_Key_Man	The TOE must manage cryptographic keys in a secure manner including generation of cryptographic keys using the TOE random number generator as source of randomness.
4	O.DAC	The TOE must control and restrict user access to the TOE protected capabilities and shielded location 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.
5	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 ⁹ . The TOE shall ensure that the data security attributes being exported are unambiguously associated with the data.

⁹ Note, the data in a sealed data blob will be protected in confidentiality but will not be protected in integrity.

#	Objective	Description
6	O.Fail_Secure	The TOE must enter a secure failure mode in the event of a failure.
7	O.General_Integ_Checks	The TOE must provide checks on system integrity and user data integrity.
8	O.I&A	The TOE must identify all users, and shall authenticate the claimed identity except the user "World" before granting a user access to the TOE facilities.
9	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 authorized 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 (except the verification of the integrity of the data within a sealed data blob).
10	O.Limit_Actions_Auth	The TOE must restrict the actions a user may perform before the TOE verifies the identity of the user. This includes requirements for physical presence of the user.
11	O.Locality	The TOE must control access to objects based on the locality of the process communicating with the TPM.
12	O.Record_Measurement	The TOE must support calculating hash values and recording the result of a measurement.
13	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.
14	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.
15	O.Reporting	The TOE must report measurement digests and attests to the authenticity of measurement digests.
16	O.Security_Attr_Mgt	The TOE must allow only authorized users to initialize and to change security attributes of objects and subjects. The management of security attributes shall support the principle of least privilege by means of role based administration and separation of duty.

#	Objective	Description
17	O.Security_Roles	The TOE must maintain security-relevant roles and association of users with those roles.
18	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.
19	O.Single_Auth	The TOE must provide a single use authentication mechanism and require re-authentication to prevent "replay" and "man-in-the-middle" attacks.
20	O.Transport_Protection	The TOE must provide the confidentiality of the payload of the commands within a transport session and the integrity of the transport log of commands.
21	O.DAA	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 Direct Anonymous Attestation Protocol.
22	O.Tamper_Resistance	The TOE must resist physical tampering of the TSF by hostile users.

5.2. Security Objectives for the Operational Environment

Table 6: 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 authorized 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 able to assert just the locality 0 or Legacy only to the TPM.
3	OE.Physical_Presence	The developer of the host platform must ensure that physical presence indicated to the TOE implies interaction by an operator and is difficult or impossible to spoof by rogue software or remote attackers.

4	OE.Int_Prot_Sealed_Blob	The IT environment must protect the integrity of sealed data blobs.
5	OE.Credential	The IT environment must create EK and AIK credentials by trustworthy procedures for the root of trust for reporting.
6	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..
7	OE.DAA	The DAA issuer must support a procedure for attestation without revealing the attestation information based on the Direct Anonymous Attestation Protocol.

5.3. Security Objectives Rationale

The table 7 provides an overview of the mapping between the security objective for the TOE and the functional security requirements.

Table 7: Security objective rationale

TOE	O.Anonymity	O.Context_Management	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Transport_Protection	O.DAA	O.Tamper_Resistance	OE.Configuration	OE.Locality	OE.Physical_Presence	OE.Int_Prot_Sealed_Blob	OE.Credential	OE.Measurement	OE.DAA
T.Compromise			X					X									X												
T.Bypass																X													
T.Export					X											X						X							
T.Hack_Crypto		X																											
T.Hack_Physical			X																		X								
T.Imperson							X	X	X	X							X						X	X					
T.Import								X																	X				
T.Insecure_State					X	X										X						X							
T.Intercept																				X									
T.Malfunction					X												X												
T.Modify			X				X	X								X													
T.Object_Attr_Change																X													
T.Replay																			X										
T.Repudiate_Transact													X																

TOE	O.Anonymity	O.Context_Management	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Atr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Transport_Protection	O.DAA	O.Tamper_Resistance	OE.Configuration	OE.Locality	OE.Physical_Presence	OE.Int_Prot_Sealed_Blob	OE.Credential	OE.Measurement	OE.DAA
T.Residual_Info														X															
OSP.Anonymity	X																												
OSP.Context_Management		X																											
OSP.Delegation				X											X														
OSP.Locality										X													X						
OSP.RT_Measurement												X															X		
OSP.RT_Reporting														X												X			
OSP.RT_Storage			X	X	X			X	X																				
OSP.Anonymous_Attestation																					X							X	
A.Configuration																						X							
A.Phys_Presence																								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 authorized to perform.

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

- O.I&A: The TOE must identify all users, and shall authenticate the claimed identity except the user "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 location 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 unauthorized actions through a defined access control policy.
- 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 unauthorized individual or user may tamper with TSF, security attributes or other data in order to bypass TOE security functions and gain unauthorized access to TOE assets.

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

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

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 initialization and management of security attributes of objects and subjects to authorized users only. The objective OE.Configuration requires the authorized user to manage these security attributes securely. Thus the object can not be exported with insecure security attributes.

T.Hack_Crypto: Cryptographic key generation or operation may be incorrectly implemented, allowing an unauthorized 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 in a secure manner including generation of cryptographic keys using the TOE random number generator as source of randomness.

T.Hack_Physical: An unauthorized individual or user of the TOE may cause unauthorized disclosure or modification of TOE assets by physically interacting with the TOE. The attacker may be a hostile user of the TOE. 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 location according to O.DAC.

T.Imperson: An unauthorized individual may impersonate an authorized user of the TOE and thereby gain access to TOE data in shielded location and protected capabilities.

T.Imperson is countered by O.I&A, O.Security_Roles, O.Import, O.Locality, OE.Locality, O.Limit_Actions_Auth and OE.Physical_Presence. These objectives prevents 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 the user "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 authorized 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 (except the verification of the integrity of the data within a sealed data blob).
- 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 includes requirements for physical presence of the user to restrict the actions a user may perform before the TOE verifies the identity of the user.
- OE.Physical_Presence requires the developer of the host platform to ensure that physical presence indicated to the TOE implies direct interaction by a operator and is difficult or impossible to spoof by rogue software or remote attackers.

T.Import: A user or attacker may import data without security attributes or with erroneous security attributes, causing key ownership and authorization to be uncertain or erroneous 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 authorized 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 (except the verification of the integrity of the data within a sealed data blob). The integrity of the data in a sealed data blob (which is not protected by the TOE itself) shall be protected by the IT environment as stated in OE.Int_Prot_Sealed_Blob.

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_Integr_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 authorized users to initialize 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_Integr_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.Transport_Protection, which states: The TOE must provide the confidentiality of the payload of the commands within a transport session and the integrity of the transport log of commands.

T.Malfunction: TOE assets may be modified or disclosed to an unauthorized 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 an 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 unauthorized modification or destruction of the information by an attacker.

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

- O.Lim_Action_Auth: The TOE must restrict the actions a user may perform before the TOE verifies the identity of the user. This includes requirements for physical presence of the user.
- O.I&A: The TOE must identify all users, and shall authenticate the claimed identity except the user "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 location 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 unauthorized 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 authorized users to initialize and to change security attributes of objects and subjects.

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

T.Replay is directly countered by O.Single_Auth, which states: The TOE must provide a single use 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 authorized 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.

OSP.Anonymity: The TOE shall support authorized users to hide the platform or attestation identity.

OSP.Anonymity is implemented by the O.Anonymity, which states: The TOE must allow the user authenticated by operatorAuth and the user "World" under physical presence temporarily to deactivate the TPM and to hide the TPM attestation identity during a user session.

OSP.Context_Management: A resource manager shall be able to secure caching of resources without knowledge or assistance from the application that loaded the resource.

The OSP.Context_Management is implemented by the O.Context_Management, which states: The TOE must ensure a secure wrapping of a resource (except EK and SRK) 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.

OSP.Delegation: The TPM support multiple trusted processes obeying the principle of least privilege by means of role based administration and separation of duty.

The OSP.Delegation 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 location 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 authorized users to initialize 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.

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 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 OSP.Locality 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 able to assert just the locality 0 or Legacy only to the TPM.

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) provided to the TPM by other parts of the root of trust for measurement.

The OSP.RT_Measurement is implemented by the TOE and a platform part of the root of trust for measurement. This implementation implies two security objectives.

- The objective **O.Record_Measurement**, which describes the responsibility of the TOE: The TOE must support calculating hash values and recording the result of a measurement.
- The objective for the environment **OE.Measurement**, which 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..

OSP.RT_Reporting: The TPM as root of trust for reporting attests the authenticity of measurement digests based on trusted platform identities by means of digital signatures with the certified AIK.

The OSP.RT_Reporting is implemented by the objectives

- **O.Reporting**, which states: The TOE must report measurement digests and attests to the authenticity of measurement digests.
- **OE.Credentials**, which addresses trustworthy procedures for creation of EK and AIK credentials for root of trust for reporting.

OSP.RT_Storage: The TPM as root of trust for storage protects the keys and data entrusted to the TPM in confidentiality and integrity.

The OSP.RT_Storage 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 in a secure manner including generation of cryptographic keys using the TOE random number generator as source of randomness. 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 authorized 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 (except the verification of the integrity of the data within a sealed data blob). 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 the user "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 location 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.

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

The **OSP.Anonymous_Attestation** is implemented by the security objectives **O.DAA** for the TOE and **OE.DAA** 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.DAA:** 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 Direct Anonymous Attestation Protocol.
- **OE.DAA:** The DAA issuer must support a procedure for attestation without revealing the attestation information based on the Direct Anonymous Attestation Protocol.

A.Configuration: The TOE will be properly installed and configured.

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 authorized user.

A.Physical_Presence: The Host Platform's trusted processes assert physical presence of local operator to the TPM.

The **A.Physical_Presence** is directly covered by the objective for the TOE environment **OE.Physical_Presence**, which states: The developer of the host platform must ensure that

physical presence indicated to the TOE implies interaction by an operator and is difficult or impossible to spoof by rogue software or remote attackers.

6. Security Requirements

6.1. Security Functional Requirements for the TOE

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

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted in the changed element in **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, a separate attachment indicates the words that were removed.

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 *italic 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 *italicized*.

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 *italic 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 *italicized*.

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.

6.1.1. General SFR

Security management

FMT_SMR.1 Security roles

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

FMT_SMR.1.1 The TSF shall maintain the roles

- (1) *TPM owner*,
- (2) *Entity owner*,
- (3) *Delegated entity*,
- (4) *Entity user*,

(5) *User using operatorAuth,*

(6) *“World”*¹⁰.

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

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

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

(1) *Management of the TPM modes of operation,*

(2) *Management of Delegation Tables and Family Tables,*

(3) *Management of security attributes of keys,*

(4) *Management of security attributes of PCR,*

(5) *Management of security attributes of NV storage areas,*

(6) *Management of security attributes of monotonic counters,*

(7) *Reset the Action Flag of TPM dictionary attack mitigation mechanism,*

(8) *[assignment: list of additional management functions to be provided by the TSF]*¹¹.

Application note 2: The ST writer shall perform the missing operation in the element FMT_SMF.1.1 to assign additional management functions to be provided by the TSF as provided by the TOE. This assignment may be empty.

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*].

¹⁰ [assignment: *the authorised identified roles*]

¹¹ [assignment: *list of management functions to be provided by the TSF*]

FPT_TDC.1 Inter-TSF basic TSF data consistency

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

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret [assignment: *list of TSF data types*] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall *use roles defined in [6] and [7]*¹² when interpreting the TSF data from another trusted IT product.

6.1.2. Cryptographic support

The cryptographic key generation by means of the TPM random number generator is an important security feature of the TPM. The random number generator shall be used for cryptographic key generation and other cryptographic mechanisms.

FCS_CKM.1 Cryptographic key generation

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

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *RSA key generator*¹³ and specified cryptographic key sizes *RSA 512, 1024, 2048*¹⁴ that meet the following: *P1363 [12]*¹⁵.

Application note 3: Note that the TPM main specification requires the TOE to implement generation of asymmetric key pairs [5], sec.4.2.3.1. The generate function is a protected capability and the private key is held in a shielded location. The implementation of the generate function must be in accordance with P1363. If the TPM support the generation for other asymmetric algorithm the ST writer shall iterate the component FCS_CKM.1.

FCS_RNG.1 Random number generation

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

¹² [assignment: *list of interpretation rules to be applied by the TSF*]

¹³ [assignment: *cryptographic key generation algorithm*]

¹⁴ [assignment: *cryptographic key sizes*]

¹⁵ [assignment: *list of standards*]

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

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

Application note 4: A physical random number generator (RNG) produces the random number by a noise source based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes like human interaction (e.g. key strokes, mouse movement). A deterministic RNG uses a random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs. The list of security capabilities may include tests of the internal noise source in case of physical RNG

Application note 5: The TPM main specification [5], sec 4.2, describes the RNG as component of the TPM. The TOE uses the RNG for creation of nonce (cf. [5], sec. 4.2.3.2), key generation and randomness in signatures [5], sec. 4.2.5.2. The TPM specification, part 2, [6], sec. 5.5, defines the nonce value as being of 20 bytes provided by the RNG.

The ST writer shall perform the operation depending on the type and security capabilities of the TOE random number generator in the element FCS_RNG.1.1. The ST writer may consider as security capability of the RNG the features described in [5], sec. 4.2.5, or TOE specific features e.g. a total failure test of the random source or internal quality test of the random numbers. The ST writer shall perform the operation depending on the quality and of the TOE random number generator in the element FCS_RNG.1.2. This assignment shall ensure the quality of the random numbers according to the resistance to guessing attacks performed by an attacker possessing Moderate attack potential as required by AVA_VAN.4 or higher attack potential if AVA_VAN.5 is included in the ST. The ST writer may consider min-entropy¹⁶ or Shannon entropy¹⁷ of independent bits as quality metric of the random numbers generated by a physical RNG. E.g. the nonce shall have at least 64 bit min-entropy to resist attacks with moderate attack potential. A deterministic RNG should implement the features described in [5], sec. 4.2.5 and ensure of the seed. The seed shall contain sufficient entropy not only for generation of nonce but also for creation of cryptographic keys which may require more than 64 bit min-entropy.

¹⁶ Min-entropy of a discrete probabilistic distribution $\{p_1, K, p_n\}$ is defined as $-\log_2 \left(\max_i \{p_i\} \right)$.

¹⁷ Shannon entropy of a discrete probabilistic distribution $\{p_1, K, p_n\}$ is defined as $-\sum_i p_i \log_2 p_i$.

FCS_CKM.4 Cryptographic key destruction

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

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

FCS_COP.1/SHA Cryptographic operation

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

FCS_COP.1.1/SHA The TSF shall perform *hash calculation*¹⁸ in accordance with a specified cryptographic algorithm *SHA-1*¹⁹ and cryptographic key sizes *not applicable*²⁰ that meet the following: *FIPS 180-2 [13]*²¹.

FCS_COP.1/HMAC Cryptographic operation

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

FCS_COP.1.1/HMAC The TSF shall perform *HMAC calculation and verification*²² in accordance with a specified cryptographic algorithm *HMAC*²³ *SHA-1* and cryptographic key sizes *160 bits*²⁴ that meet the following: *RFC2104 [14]*²⁵ and *FIPS180-2 [13]*.

¹⁸ [assignment: *list of cryptographic operations*]

¹⁹ [assignment: *cryptographic algorithm*]

²⁰ [assignment: *cryptographic key sizes*]

²¹ [assignment: *list of standards*]

²² [assignment: *list of cryptographic operations*]

²³ [assignment: *cryptographic algorithm*]

²⁴ [assignment: *cryptographic key sizes*]

²⁵ [assignment: *list of standards*]

FCS_COP.1/RSA_Sig Cryptographic operation

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

FCS_COP.1.1/RSA_Sig The TSF shall perform *signature generation and signature verification*²⁶ in accordance with a specified cryptographic algorithm *RSA*²⁷ *signature scheme [5] section [selection: 31.2.1, 31.2.2 or 31.2.3]* and cryptographic key sizes *RSA 512, 1024, 2048*²⁸ that meet the following: *PKCS#1 V2.0 [15]*²⁹.

FCS_COP.1/RSA_Enc Cryptographic operation

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

FCS_COP.1.1/RSA_Enc The TSF shall perform *encryption and decryption*³⁰ in accordance with a specified cryptographic algorithm *RSA*³¹ *encryption scheme [5] section [selection: 31.1.1, 31.1.2, 31.1.3 or 31.1.4]* and cryptographic key sizes *RSA 512, 1024, 2048*³² that meet the following: *PKCS#1 V2.0 [15]*³³.

FCS_COP.1/SymEnc Cryptographic operation

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

²⁶ [assignment: *list of cryptographic operations*]

²⁷ [assignment: *cryptographic algorithm*]

²⁸ [assignment: *cryptographic key sizes*]

²⁹ [assignment: *list of standards*]

³⁰ [assignment: *list of cryptographic operations*]

³¹ [assignment: *cryptographic algorithm*]

³² [assignment: *cryptographic key sizes*]

³³ [assignment: *list of standards*]

FCS_COP.1.1/SymEnc The TSF shall perform *symmetric encryption and decryption*³⁴ in accordance with a specified cryptographic algorithm *TPM_ALG_MGF1*³⁵ and cryptographic key sizes *variable bit length*³⁶ that meet the following: *PKCS#1 V2.0 [15] and [6]*³⁷.

Application note 6: The TPM main specification [6], sec. 4.8, lists types of cryptographic algorithms the TOE may support. This list includes symmetric cryptographic encryption / decryption algorithms DES [17], 3DES in EDE mode [18], AES-128, AES-196, AES 256 [16] and xoring of plain or cipher text with rolling nonce (TPM_ALG_XOR) or strings generated using MGF1 [15] (TPM_ALG_MGF1), which are generated according the standard *PKCS#1 V2.0 [15] and [6]* and used as a key stream. The encryption and decryption take place by “xoring of plain or cipher text” with the key stream. The security target writer shall iterate the component FCS_COP.1 to describe all symmetric encryption algorithms supported by the TSF. TPM_ALG_MGF1 must be supported by the TOE (cf. [6], sec. 4.8). E. g. TPM_ALG_MGF1 may be used to encrypt (cf. to command TPM_Sealx) or decrypt (cf. to command TPM_Unseal) sealed data blobs, where the sender and the receiver must use the same secret. The seed value of the MGF1 has different length depending on their application.

6.1.3. TPM Operational Modes

The TOE enters an operational mode defined by the flags “disable“, “deactivated” and “ownership” in the TPM_PERMANENT_FLAGS (cf. [6] sec. 7.1) after TPM_Init and TPM_Startup. The flag “deactivated” in TPM_PERMANENT_FLAGS is copied to the flag “deactivated” in the TPM_STCLEAR_FLAGS which will define the TPM operational mode related to activation during the power-on session. The flag “deactivated” in the TPM_STCLEAR_FLAGS may be set temporarily to TRUE until the next boot by means of the command TPM_SetTempDeactivated (i.e. it can not be reset to FALSE within the power-on session).

Application note 7: The figure 2 illustrates the transition between the TPM operational modes as defined in [6], ch. 17, and [7], ch. 5 and 6.

³⁴ [assignment: *list of cryptographic operations*]

³⁵ [assignment: *cryptographic algorithm*]

³⁶ [assignment: *cryptographic key sizes*]

³⁷ [assignment: *list of standards*]

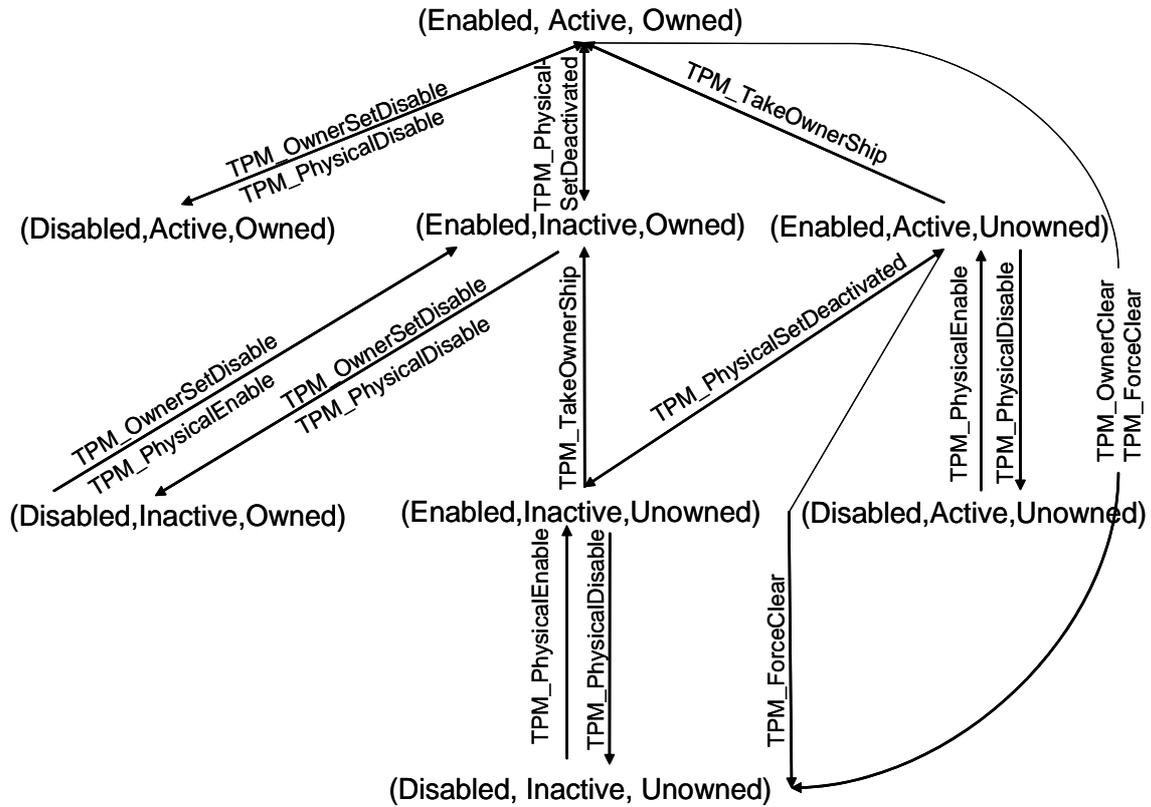


Figure 2: Transition of the TPM modes of operation
(End of the application note 7)

FDP_ACC.1/Modes Subset access control

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

FDP_ACC.1.1/Modes The TSF shall enforce the *TPM Mode Control SFP*³⁸ on all subjects, all objects and all commands³⁹.

FDP_ACF.1/Modes Security attribute based access control

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

³⁸ [assignment: *access control SFP*]

³⁹ [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

FDP_ACF.1.1/Modes The TSF shall enforce the *TPM Mode Control SFP*⁴⁰ to objects based on the following: *all subjects and all objects, flags disable, deactivated, owner and ownership*⁴¹.

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

- (1) *The TPM shall prevent the execution of a command if the TPM is disabled and the command to be executed for the operation is not available according to table 12, column Avail Disabled,*
- (2) *The TPM shall prevent the execution of a command if the TPM is permanently or temporarily inactive and the command to be executed for the operation is not available according to table 12, column Avail Deactivated,*
- (3) *The TPM shall prevent the execution of a command if the TPM is unowned and the command to be executed for the operation is not allowed according to table 12, column No Owner*⁴².

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

- (1) *The command marked with "A" in table 12, column Avail Disabled is allowed to be executed if the TPM is disabled and the underlying NV storage does not require authorization*
- (2) *The command to be executed for the operation is marked with "A" in table 12, column Avail Deactivated, is allowed to be executed if the TPM is permanently or temporarily inactive and the underlying NV storage does not require authorization*⁴³.

FDP_ACF.1.4/Modes The TSF shall explicitly deny access of subjects to objects based on the rule: *none*⁴⁴.

⁴⁰ [assignment: *access control SFP*]

⁴¹ [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*]

⁴² [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

⁴³ [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

⁴⁴ [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

Application note 8: The columns Avail Disabled, Avail Deactivated and No Owner, table 12, identify the command ordinals available for disabled, inactive or unowned TPM with “X” or “A”. All other commands are not executable. The flags “disable”, “deactivated” and “ownership” are permanent stored security attributes of the TOE relevant for all subjects and objects. The value of the flag “deactivated” may be changed temporarily.

FMT_MSA.1/Modes Management of security attributes

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

FMT_MSA.1.1/Modes The TSF shall enforce the *TPM Mode Control SFP*⁴⁵ to restrict the ability to *modify*⁴⁶ the security attributes *TPM operational mode flags disable, deactivated and ownership*⁴⁷ to *TPM owner, role using operatorAuth and user “World” under physical presence*⁴⁸ **based on the rules:**

- (1) the TPM is disabled, inactive and unowned when created,**
- (2) the TPM owner is allowed to set the TPM operational modes to disabled, inactive and unowned,**
- (3) the TPM owner is allowed to set the TPM operational modes to enabled and disabled,**
- (4) a user “World” is allowed to own an enabled and unowned TPM if the flag ownership is TRUE,**
- (5) a user “World” under physical presence is allowed to set the TPM operational modes to disabled, inactive and unowned at once,**
- (6) a user “World” under physical presence is allowed to set permanently an enabled TPM to active and inactive,**
- (7) the user “World” under physical presence is allowed to deactivate temporarily an enabled and active TPM,**
- (8) the user authenticated by operatorAuth is allowed to deactivate temporarily an enabled and active TPM,**

⁴⁵ [assignment: *access control SFP, information flow control SFP*]

⁴⁶ [selection: *change_default, query, modify, delete, [assignment: other operations]*]

⁴⁷ [assignment: *list of security attributes*]

⁴⁸ [assignment: *the authorised identified roles*]

- (9) a user “World” under physical presence is allowed to set the TPM operational modes to enabled and to disabled,**
- (10) a user is not allowed to own an disabled or owned TPM,**
- (11) a user is not allowed to activate or deactivate an disabled TPM without setting unowned at the same time.**
- (12) a user “World” under physical presence is allowed to set the the flag ownership to TRUE,**
- (13) the TPM owner is allowed to modify the flag ownership.**

Application note 9: The command TPM_SetTempDeactivated allows an user authenticated with operatorAuth or a user under physical presence to deactivate temporarily an active TPM, i.e. until next reboot which re-establishes the permanent value of the security attribute deactivated of the TPM_PERMANENT_FLAGS. The commands TPM_OwnerSetDisable, TPM_PhysicalDisable and TPM_PhysicalSetDeactivated set permanent the attributes of the TPM operational modes only. The commands TPM_OwnerClear and TPM_ForceClear delete also operatorAuth, SRK, tpmProof, delegate keys, delegate table and other objects. Note the host platform (i.e. IT environment) ensures that physical presence indicated to the TOE implies direct interaction by a person and is difficult or impossible to spoof by rogue software or remote attackers (cf. to OE.Physical_Presence).
(end of the application note)

The SFR FMT_MSA.1/PhysP describes the requirements for assertion and management of the security attribute “*physical presence*” of users. The security attribute *physical presence* is set during start-up and may be asserted by setting the PhysicalPresence flag in the TPM_STCLEAR_FLAGS to the possible values TPM_PHYSICAL_PRESENCE_PRESENT and TPM_PHYSICAL_PRESENCE_NOTPRESENT (cf. TPM spec part 2, sec. 7.2) as follows:

- *set to the default value:* by the TPM_Startup[ST_CLEAR] and TPM_Startup[ST_DEACTIVATED] (cf. [7], sec. 3.2) and by the TPM_Startup[TPM_ST_STATE] to the value, which was saved by the command TPM_SaveState (cf. [7], sec. 3.3);
- *assertion by HW:* the user (i.e. an external entity like PC platform BIOS) indicates physical presence to the TPM through a hardware pin, i.e. he sets *physical presence* to TRUE while physical presence is indicated for the current command to be executed and sets to FALSE while physical presence is not indicated to the TPM;
- *assertion by command:* the command TSC_PhysicalPresence sets *physical presence* to TRUE if TPM_PHYSICAL_PRESENCE = TPM_PHYSICAL_PRESENCE_PRESENT and *physical presence* to FALSE if TPM_PHYSICAL_PRESENCE = TPM_PHYSICAL_PRESENCE_NOTPRESENT (cf. TPM spec part 2 [6], sec. 4.9, and part 3 [7], sec. 6.6, for details).

The management of the security attribute “physical presence” is controlled by the command TSC_PhysicalPresence. These functions are defined as follows (cf. TPM spec part 2 [6], sec. 4.9, and part 3 [7], sec. 6.6, for details)

- *enable HW setting*: the command TSC_PhysicalPresence with TPM_PHYSICAL_PRESENCE = TPM_PHYSICAL_PRESENCE_HW_ENABLE - sets physicalPresenceHWEEnable in the TPM_PERMANENT_FLAGS to TRUE, which allows the hardware Physical Presence input to modify the state of the PhysicalPresence flags,
- *disable HW setting*: the command TSC_PhysicalPresence with TPM_PHYSICAL_PRESENCE = TPM_PHYSICAL_PRESENCE_HW_DISABLE - sets physicalPresenceHWEEnable in the TPM_PERMANENT_FLAGS to FALSE, which disallows the hardware Physical Presence input to modify the state of the PhysicalPresence flags,
- *enable SW setting*: the command TSC_PhysicalPresence with TPM_PHYSICAL_PRESENCE = TPM_PHYSICAL_PRESENCE_CMD_ENABLE - sets physicalPresenceCMDEnable in the TPM_PERMANENT_FLAGS to TRUE, which allows the TSC_PhysicalPresence command to modify the state of the PhysicalPresence flags,
- *disable SW setting*: the command TSC_PhysicalPresence with TPM_PHYSICAL_PRESENCE = TPM_PHYSICAL_PRESENCE_CMD_DISABLE - sets physicalPresenceCMDEnable in the TPM_PERMANENT_FLAGS to FALSE, which disallows the TSC_PhysicalPresence command to modify the state of the PhysicalPresence flags
- *locking permanently*: the command TSC_PhysicalPresence with TPM_PHYSICAL_PRESENCE = TPM_PHYSICAL_PRESENCE_LIFETIME_LOCK - sets physicalPresenceLifetimeLock in the TPM_PERMANENT_FLAGS to TRUE, which permanently locks the state of the Physical Presence permanent flags.
- *locking temporarily*: the command TSC_PhysicalPresence with TPM_PHYSICAL_PRESENCE = TPM_PHYSICAL_PRESENCE_LOCK - sets PhysicalPresenceLock in the TPM_STCLEAR_FLAGS to TRUE, which disables the ability to assert Physical Presence until TPM_Startup(ST_Clear) is executed, and sets the physicalPresence assertion in the TPM_STCLEAR_FLAGS to FALSE.

FMT_MSA.1/PhysP Management of security attributes

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

FMT_MSA.1.1/PhysP The TSF shall enforce the *TPM Mode Control SFP, Delegation SFP, Key Management SFP, NVS SFP*⁴⁹ to restrict the ability to *set to the default value, assert by HW, assert by command, enable HW setting, disable HW setting, enable SW setting, disable SW setting, locking temporarily, locking*

⁴⁹ [assignment: *access control SFP, information flow control SFP*]

*permanently*⁵⁰ the security attributes *physical presence*⁵¹ to user "World"⁵²
based on the additional rules:

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

Application note 10: The indication of a physical presence at the platform either provides another indication of platform ownership (i.e. an operator is direct interacting with the platform / TPM, cf. TPM spec part 1, ch. 10) or a mechanism to ensure that the execution of the command is not the result of a remote software process. The physical presence is required for the execution of privileged commands as defined in the ordinal table, column "physical presence" to enforce the listed *TPM Mode Control SFP* (cf. FDP_MSA.1/Modes), *Delegation SFP*, *Key Management SFP*, *NVS SFP* (cf. to the respective SFR FDP_ACC.1 and FDP_ACF.1).

- to the default value by the `TPM_Startup[ST_CLEAR]` and `TPM_Startup[ST_DEACTIVATED]` (cf. TPM spec part 3, sec. 3.2)
- by the `TPM_Startup[TPM_ST_STATE]` to the value, which was saved by the command `TPM_SaveState` (cf. TPM spec part 3, sec. 3.2)

The security attribute *physical presence* may be asserted by setting `PhysicalPresence` flag in the `TPM_STCLEAR_FLAGS` to the possible values `TPM_PHYSICAL_PRESENCE_PRESENT` and `TPM_PHYSICAL_PRESENCE_NOTPRESENT` (cf. TPM spec part 2, sec. 7.2) by means of

- hardware `Physical Presence` input while physical presence is indicated, and
- `TPM_Startup` and the command `TSC_PhysicalPresence` setting temporarily for more than one command.

The assertion of security attribute *physical presence* is managed by means of the command `TSC_PhysicalPresence` by

- permanent setting over the life time in the `TPM_PERMANENT_FLAGS` (cf. TPM spec part 2, sec. 7.1) and
- temporarily settings for a TPM power-on session in the `TPM_STCLEAR_FLAGS` (cf. TPM spec part 2, sec. 7.2).

6.1.4. Identification, Authentication and Binding

⁵⁰ [selection: *change_default*, *query*, *modify*, *delete*, [assignment: *other operations*]]

⁵¹ [assignment: *list of security attributes*]

⁵² [assignment: *the authorised identified roles*]

Application note 11: The TPM identification and authentication capability is used to authenticate an entity owner and to authorize use of an entity. The basic premise is to prove knowledge of a shared secret. This shared secret AuthData is assigned to the object as security attribute for the identification of the object owner and as TSF data for authentication data of the object owner. Note that the TCG Main Specification document refers to the identification and authentication process and this data as *authorization*.

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

The identification and authentication protocols use a random nonce. This requires that a nonce from one side be in use only for a message and its reply. For instance, the TPM would create a nonce and send that on a reply. The requestor would receive that nonce (nonceOdd) and then include it in the next request. The TPM would validate that the correct nonce was in the request and then create a new nonce for the reply (nonceEven).

FMT_MTD.1/AuthData Management of TSF data

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

FMT_MTD.1.1/AuthData The TSF shall restrict the ability to *modify and create*⁵³ the *authentication data*⁵⁴ to *TPM Owner, user under physical presence and Entity Owner*⁵⁵ **based on the rules:**

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

⁵³ [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

⁵⁴ [assignment: *list of TSF data*]

⁵⁵ [assignment: *the authorised identified roles*]

- (5) The user under physical presence modifies the authentication data `operatorAuth` by successful execution of the command `TPM_SetOperatorAuth`.
- (6) The Entity Owner modifies the authentication data for the owned object by successful execution of the commands `TPM_ChangeAuth`.
- (7) The Entity Owner modifies the authentication data for the owned object by successful execution of the commands `TPM_ChangeAuthAsymStart` and `TPM_ChangeAuthAsymFinish`.

Application note 12: The Authorization-Data Insertion Protocol (ADIP) is described in [5], sec. 13.5.

FMT_MTD.1/Deleg Management of TSF data

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

FMT_MTD.1.1/Deleg The TSF shall restrict the ability to *modify and create*⁵⁶ the *authentication data of a delegation blob*⁵⁷ to TPM Owner and authorized users⁵⁸ based on the rules:

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

Application note 13: The delegation blob is returned by the command `TPM_Delegate_CreateOwnerDelegation`, parameter 5.

⁵⁶ [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

⁵⁷ [assignment: *list of TSF data*]

⁵⁸ [assignment: *the authorised identified roles*]

FIA_UID.1 Timing of identification

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

FIA_UID.1.1 The TSF shall allow

- (1) *to execute commands indicated in table 12 column RQU as not requesting authentication,*
- (2) *accessing objects where entity owner has given the user "World" access based on the value of TPM_AUTH_DATA_USAGE,*
- (3) *[assignment: list of other TSF-mediated actions]*⁵⁹

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

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

FIA_UAU.1 Timing of authentication

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

FIA_UAU.1.1 The TSF shall allow

- (1) *to execute commands indicated in table 12 column RQU as not requesting authentication,*
- (2) *accessing objects where entity owner has given the user "World" access based on the value of TPM_AUTH_DATA_USAGE,*
- (3) *[assignment: list of other TSF-mediated actions]*⁶⁰

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 14: The list of commands which may be executed without authorization is defined in [6], chapter 17 (see table 12 in 7.1 Ordinal table). The ST writer shall perform the assignment to define other TSF-mediated actions the user may perform without identification in the element FIA_UID.1.1 and FIA_UAU.1. This may be a platform specific subject e.g. PC

⁵⁹ [assignment: *list of TSF-mediated actions*]

⁶⁰ [assignment: *list of TSF mediated actions*]

client specific TPM will allow users with Locality 4 to bind to a HASH session running the commands TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END. Another example is optional commands defined in the TPM main specification like TPM_CreateRevokableEK (cf. [6], chapter 17, for a list of optional commands).

FIA_UAU.4 Single-use authentication

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

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to

- (1) *OIAP authorization session,*
- (2) *OSAP authorization session,*
- (3) *DSAP authorization session*
- (4) *Transport session*⁶¹.

FIA_UAU.5 Multiple authentication mechanisms

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

FIA_UAU.5.1 The TSF shall provide

- (1) *OIAP authorization session,*
- (2) *OSAP authorization session,*
- (3) *DSAP authorization session,*
- (4) *Transport session,*
- (5) *Commands which require authorization and are executed outside a authorization session*⁶².

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [assignment: *rules describing how the multiple authentication mechanisms provide authentication*].

Application note 15: The TPM main specification, part 1 [5], section 8.1) requires vendor specific mechanisms to manage authentication failure handling to mitigate dictionary attacks. Note this version [5] [6] [7] of the TPM specification does not specify the particular actions to be used to react on detected dictionary attacks (e.g. locking out the TPM after a certain

⁶¹ [assignment: *identified authentication mechanism(s)*]

⁶² [assignment: *list of multiple authentication mechanisms*]

number of failures, forcing a reboot under some combination of failures, or requiring specific actions on the part of some actors after an attack has been detected). This PP describes the behavior of the TPM dictionary attack mitigation mechanism using two objects: (i) a Action Flag indicating the activation (TRUE) or deactivation (FALSE) of the TPM dictionary attack mitigation actions, and (ii) the flag *TPM_STCLEAR_DATA -> disableResetLock* which is TRUE for the timeout period and set to FALSE when the timeout period expires (cf. [7], section 9.3). The TPM activates the TPM dictionary attack mitigation actions (by setting the Activation Flag to TRUE) when dictionary attacks are detected (cf. FIA_AFL.1.2 (1)). The TPM dictionary attack mitigation actions include time out controlled by *TPM_STCLEAR_DATA -> disableResetLock* flag. The TPM shall reject authentication attempts if the value *TPM_STCLEAR_DATA -> disableResetLock* is TRUE as described by FIA_TBR.1 (cf. [6], section 7.5). The TPM main specification, part 3 [7], section 9.3, specifies the command *TPM_ResetLockValue* that allows the TPM Owner to reset the TPM dictionary attack mitigation mechanism (described here by Action Flag set to FALSE). This PP description of the security functional requirements does not enforce any vendor specific implementation the flags of the TPM dictionary attack mitigation mechanism.

FIA_UAU.6 Re-authenticating

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

FIA_UAU.6.1 The TSF shall re-authenticate the user under the condition *the user sent a command that requires authentication within a session*⁶³.

FIA_AFL.1 Authentication failure handling

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

FIA_AFL.1.1 The TSF shall detect when *[assignment: positive integer number]*⁶⁴ unsuccessful authentication attempts occur related to *authentication attempts for the same user*⁶⁵.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *[selection: met, surpassed]*, the TSF shall

- (1) *Set the Action Flag to TRUE,*
- (2) *[assignment: list of other actions]*⁶⁶.

⁶³ [assignment: *list of conditions under which re-authentication is required*]

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

⁶⁵ [assignment: *list of authentication events*]

⁶⁶ [assignment: *list of actions*]

FMT_MTD.1/Lock Management of TSF data

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

FMT_MTD.1.1/Lock The TSF shall restrict the ability to *reset to FALSE*⁶⁷ the *Action Flag of TPM dictionary attack mitigation mechanism*⁶⁸ to the *TPM Owner and Delegated Entity*⁶⁹.

Application note 16: The ST writer shall perform the operation in the elements FIA_AFL.1.1 and FIA_AFL.1.2 to describe the TOE security requirements for countermeasures against dictionary attacks (cf. TPM main specification, part 1 [5], section 8.1). The TPM dictionary attack mitigation mechanism to manage authentication failure handling is vendor specific. If dictionary attack are detected the Action Flag is set to TRUE. The TPM Owner may reset the Action Flag of the TPM dictionary attack mitigation mechanism to FALSE by successfully executing the command TPM_ResetLockValue. The TPM_ResetLockValue must be allowed to run exactly once while the TPM is locked up.

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) *authData*,
- (2) *locality*,
- (3) *physical presence*,
- (4) *authorization handle and shared secret if the subject is a OSAP or DSAP session*,
- (5) *authorization associated with the delegation blob if the subject is a DSAP session*⁷⁰.

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 shared secret is associated with the authorization gained by the user providing the AuthData for the entity identified in the TPM_OSAP command establishing the OSAP session*,

⁶⁷ [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

⁶⁸ [assignment: *list of TSF data*]

⁶⁹ [assignment: *the authorised identified roles*]

⁷⁰ [assignment: *list of user security attributes*]

- (2) *the shared secret is associated with the authorization gained by the user providing the AuthData and the delegation blob for establishing the DSAP session,*
- (3) *the present value of the users locality is assigned to the command executed by this user,*
- (4) *the physical presence of the user is assigned to the command executed by that user⁷¹.*

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 set the security attributes of the subject TPM Owner to values defined by the command TPM_OwnerClear*
 - (a) *if the subject executing the command TPM_OwnerClear is bound to the TPM owner and all command parameters and the security attribute DisableOwnerClear are FALSE,*
 - (b) *if the subject with physical presence is executing the command TPM_ForceClear and the security attribute disableForceClear is FALSE.*
- (2) *The TSF shall delete the shared secret for the authorization of the OSAP session if the user executes the command TPM_Reset.*
- (3) *The TSF shall delete the shared secret for the authorization of the OSAP session and DSAP session if*
 - (a) *the user executes the command TPM_FlushSpecific or TPM_Terminate_Handle,*
 - (b) *the user clears the TPM Owner by executing the command TPM_OwnerClear or TPM_ForceClear,*
 - (c) *the user is the TPM owner and executes the command TPM_ChangeAuthOwner,*
 - (d) *any of the following commands are executed:*
 - i. *TPM_Delegate_Manage*
 - ii. *TPM_Delegate_CreateOwnerDelegation with Increment==TRUE*
 - iii. *TPM_Delegate_LoadOwnerDelegation.*
- (4) *The TSF shall delete enforced by the user the shared secret for the authorization of all OSAP sessions associated with the counter by executing the command TPM_ReleaseCounter or TPM_ReleaseCounterOwner,*

⁷¹ [assignment: rules for the initial association of attributes]

- (5) *The TSF shall delete enforced by the user the shared secret for the authorization of the session if the user sets the continueUse flag to FALSE in the command within an OSAP or DSAP session,*
- (6) *The TSF shall delete automatically the shared secret for the authorization of the OSAP session and DSAP session acting on the behalf of users after*
 - (a) *the session executes a command that returns an error,*
 - (b) *the session uses a resource evicted from the TOE or otherwise invalidated,*
 - (c) *the session executes any command for which the shared secret is used to encrypt an input parameter (TPM_ENCAUTH)⁷².*

Application note 17: The term “terminate a session” removes the authorization of the user bind to the subject, deletes the shared secret. Unfortunately there is a contradiction in the TPM specification. The description of the command TPM_Reset in [7], sec. 27.5, specifies that this deprecated command is not upgraded to affect any other TPM entity in specification version 1.2. Since TPM_DSAP is a 1.2 command, a DSAP session must not be cleared by the TPM_Reset command. However, the definition of TPM_DSAP (section 18.3, description 7.f.) states that the session must be explicitly terminated with continueAuth, TPM_Reset or TPM_FlushSpecific⁷³. The TSF may but is not required to delete the shared secret for the authorization of the DSAP session if the user executes the command TPM_Reset.

6.1.5. Data Protection and Privacy

FDP_RIP.1 Subset residual information protection

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

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource from*⁷⁴ the following objects: *any object*⁷⁵.

6.1.5.1. Delegation

⁷² [assignment: *rules for the changing of attributes*]

⁷³ The TPM specification version 0.104 correct this inconsistency.

⁷⁴ [selection: *allocation of the resource to, deallocation of the resource from*]

⁷⁵ [assignment: *list of objects*]

FDP_ACC.1/Deleg Subset access control

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

FDP_ACC.1.1/Deleg The TSF shall enforce the *Delegation SFP*⁷⁶ on *Delegated Entities, user data and commands*⁷⁷.

FDP_ACF.1/Deleg Security attribute based access control

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

FDP_ACF.1.1/Deleg The TSF shall enforce the *Delegation SFP*⁷⁸ to objects based on the following: *Delegated Entities and commands with the delegated permission defined in the delegation table row, locality, pcrInfo and key handle of the key in the Delegation owner blob*⁷⁹.

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

- (1) *The TSF shall disallow the execution of a command in a DSAP session if the permission of this command is not set in the delegation table row in the Delegation owner blob used for the DSAP session,*
- (2) *The TSF shall disallow the execution of a command in a DSAP session if the PCR_SELECTION of the DSAP session is not NULL and the pcrInfo of the DSAP session does not match the current PCR value of the PCR_SELECTION and locality*⁸⁰.

FDP_ACF.1.3/Deleg 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*].

⁷⁶ [assignment: *access control SFP*]

⁷⁷ [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

⁷⁸ [assignment: *access control SFP*]

⁷⁹ [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*]

⁸⁰ [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

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

Application note 18: The TPM_DELEGATE_OWNER_BLOB, TPM_ET_DEL_ROW or TPM_DELEGATE_KEY_BLOB used for the TPM_DSAP command define security attributes of the DSAP session: (i) the delegated permission defined in the delegation table row (cf. [6], sec. 20.2), (ii) the pcrInfo (cf. [5], sec. 29.9) and – in case of TPM_DELEGATE_KEY_BLOB – the key handle of the key under delegation access rights (cf. [7], sec. 18.3). For TPM version 1.2 the pcrInfo uses the TPM_PCR_INFO_LONG or TPM_PCR_INFO_SHORT structure containing the PCR values and the locality.

FMT_MSA.1/DFT Management of security attributes

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

FMT_MSA.1.1/DFT The TSF shall enforce the *Delegation SFP*⁸¹ to restrict the ability to *modify, to delete, to enable, to disable and to create*⁸² the security attributes *Family table*⁸³ to

- (1) *TPM owner,*
- (2) *User under physical presence if*
 - (a) ***the opCode is TPM_FAMILY_CREATE,***
 - (b) ***DisableForceClear is FALSE and***
 - (c) ***TPM_Delegate_Admin_Lock is false***⁸⁴.

Application note 19: The operation of the family table by means command TPM_Delegate_Manage is defined by the operation code as described in [7], sec. 19.1. The CC term “delete” is used for the TCG term “invalidate”.

FMT_MSA.1/DT Management of security attributes

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

⁸¹ [assignment: *access control SFP, information flow control SFP*]

⁸² [selection: *change_default, query, modify, delete, [assignment: other operations]*]

⁸³ [assignment: *list of security attributes*]

⁸⁴ [assignment: *the authorised identified roles*]

FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1/DT The TSF shall enforce the *Delegation SFP*⁸⁵ to restrict the ability to *query, modify, create*⁸⁶ the security attributes *Delegation table*⁸⁷ to

- (1) *TPM owner,*
- (2) *User "World" if the TPM owner is not installed and max NV writes without an owner is not exceeded and TPM_Delegate_Admin_Lock is false*⁸⁸.

Application note 20: A Delegation blob contains a Delegation table row in encrypted form. Note the command `TPM_Delegate_VerifyDelegation` interprets a Delegate blob and returns success or failure, depending on whether the blob is currently valid. The delegate blob is NOT loaded into the TPM and the Delegation table is not modified.

FMT_MSA.3/Deleg Static attribute initialisation

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

FMT_MSA.3.1/Deleg The TSF shall enforce the *Delegation SFP*⁸⁹ to provide *permissive*⁹⁰ default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Deleg The TSF shall allow the *TPM owner*⁹¹ to specify alternative initial values to override the default values when an object or information is created.

6.1.5.2. Key management

The TOE associates security attributes with key objects defining access conditions checked during the command execution:

- `TPM_AUTH_DATA_USAGE`, cf. [6], sec. 5.9, defines whether the authorization of the subject is necessary and `authData` define the authorization data to be presented for use of this key known

⁸⁵ [assignment: *access control SFP, information flow control SFP*]

⁸⁶ [selection: *change_default, query, modify, delete, [assignment: other operations]*]

⁸⁷ [assignment: *list of security attributes*]

⁸⁸ [assignment: *the authorised identified roles*]

⁸⁹ [assignment: *access control SFP, information flow control SFP*]

⁹⁰ [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

⁹¹ [assignment: *the authorised identified roles*]

- (b) *SRK with the SFR-related security attribute `disableOwnerClear` and `disableForceClear` of the TOE,*
- (c) *User keys with security attributes `authDataUsage`, `keyUsage`, `keyFlags`, and `OwnerEvict`,*
- (d) *Wrapped Key Blob with the security attributes `keyUsage`, `keyFlags`, `algorithmParms` and `pcrInfo`⁹⁵.*

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

- (1) *The user "World" is allowed to create an EK if the EK does not exist already.*
- (2) *The user "World" is allowed to read the public part of an EK if the TOE is unowned.*
- (3) *The TPM owner is allowed to read the public part of an EK.*
- (4) *The user "World" is allowed to create an SRK if the ownership flag is TRUE.*
- (5) *The TPM owner is allowed to delete an SRK if the `disableOwnerClear` flag is FALSE.*
- (6) *The user "World" under physical presence is allowed to delete an SRK if the `disableForceClear` flag is FALSE.*
- (7) *The user authenticated as TPM owner and the owner of the SRK is allowed to generate an AIK.*
- (8) *The TPM owner is allowed to activate the AIK if the imported blob is a `TPM_EK_BLOB` structure and the actual state meets the identified PCR values and the locality.*
- (9) *The TPM owner is allowed to use the AIK for signing audit data, quoted data, or a tick stamped blob.*
- (10) *The entity owner of a key with the security attribute `keyUsage`, `TPM_KEY_STORAGE = TRUE`, is allowed to generate an User Key and export this User key wrapped with the key he owns except this entity owner is not the TPM owner and the key to generated is an AIK.*
- (11) *The Entity owner of the key to be used for import of Wrapped Key Blob is allowed to import a User key in a Wrapped Key Blob if the security attribute `keyUsage`, `TPM_KEY_STORAGE = TRUE`, of the import key is set.*

⁹⁵ [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*]

- (12) *The entity owner is not allowed to use a User key if at least one of the following conditions is met:*
- (a) *the security attribute `authDataUsage` of the User Key object for access does not match the authentication status of the subject,*
 - (b) *the security attribute `usageAuth` of the User Key object for access does not match the authentication data used by the user bound to the subject,*
 - (c) *the security attributes `keyUsage` or `algorithmParms` or `keyFlags` of the User Key object does not allow use of the command to be executed,*
 - (d) *the security attribute `PCRInfo` of the User Key object does not allow to use of the object in the current state of the identified PCR and locality.*
- (13) *The TPM owner is allowed to delete a User key if the security attribute `OwnerEvict`, `OwnerEvict = FALSE`⁹⁶.*

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

Application note 21: The operation “read EK” address reading of the public key component of the EK (cf. table 1).

FMT_MSA.1/KeyMan Management of security attributes

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

FMT_MSA.1.1/KeyMan The TSF shall enforce the *Key Management SFP*⁹⁷ to restrict the ability to assign the initial value⁹⁸ the security attributes

- (1) *srkParams*⁹⁹ **of the SRK** to user “World”¹⁰⁰,

⁹⁶ [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

⁹⁷ [assignment: *access control SFP, information flow control SFP*]

⁹⁸ [selection: *change_default, query, modify, delete, [assignment: other operations]*]

(2) *authDataUsage, usageAuth, keyUsage, algorithmParms, keyFlags and PCRInfo*¹⁰¹ associated with the generated User key to the entity owner¹⁰².

Application note 22: The element FMT_MSA.1.1/KeyMan is refined to list the security attributes of objects and the roles allowed to change the initial value of the security attributes. Note, srkParams are set using the command TPM_TakeOwnership which defines the authentication data of the TPM owner (i.e. ownerAuth) and the srkParams.

FMT_MSA.1/KEvi Management of security attributes

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

FMT_MSA.1.1/KEvi The TSF shall enforce the *Key Management SFP*¹⁰³ to restrict the ability to *modify*¹⁰⁴ the security attributes *TPM_KEY_CONTROL_OWNER_EVICT*¹⁰⁵ **of a loaded key** to the *Entity owner*¹⁰⁶.

FMT_MSA.3/KeyMan Static attribute initialisation

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

FMT_MSA.3.1/KeyMan The TSF shall enforce the *Key Management SFP*¹⁰⁷ to provide *restrictive*¹⁰⁸ default values for security attributes that are used to enforce the SFP.

⁹⁹ [assignment: *list of security attributes*]

¹⁰⁰ [assignment: *the authorised identified roles*]

¹⁰¹ [assignment: *list of security attributes*]

¹⁰² [assignment: *the authorised identified roles*]

¹⁰³ [assignment: *access control SFP, information flow control SFP*]

¹⁰⁴ [selection: *change_default, query, modify, delete, [assignment: other operations]*]

¹⁰⁵ [assignment: *list of security attributes*]

¹⁰⁶ [assignment: *the authorised identified roles*]

¹⁰⁷ [assignment: *access control SFP, information flow control SFP*]

¹⁰⁸ [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

FMT_MSA.3.2/KeyMan The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.

Key Migration

Application note 23: The TOE support migration keys and certified migration keys for key migration. The concept of key migration is described in [5], chapter 37. The objects and security attributes identified in FDP_ACF.1.1/MigK are described in [7], chapter 11 Migration, and [6], sec. 4.2 TPM_PAYLOAD_TYPE and chapter 10 TPM_KEY complex.

FDP_ACC.1/MigK Subset access control

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

FDP_ACC.1.1/Mig The TSF shall enforce the *Key Migration SFP*¹⁰⁹ on

- (1) *Subjects: TPM owner, Entity owner;*
- (2) *Objects: User key, Wrapped Key Blob, Migration Key Blob, Certified Migration Key Blob;*
- (3) *Operations: commands TPM_CreateMigrationBlob, TPM_CMK_CreateKey, TPM_CMK_CreateBlob, TPM_CMK_ConvertMigration, TPM_ConvertMigrationBlob, TPM_MigrateKey*¹¹⁰.

FDP_ACF.1/MigK Security attribute based access control

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

FDP_ACF.1.1/MigK The TSF shall enforce the *Key Migration SFP*¹¹¹ to objects based on the following:

- (1) *Subjects: TPM owner, Entity owner of the key with security attributes restrictDelegate and migrationScheme,*
- (2) *Objects:*
 - (a) *User key with security attribute migratable,*
 - (b) *Wrapped Key Blob with the security attribute payload type,*
 - (c) *Migration Key Blob with the security attribute payload type,*

¹⁰⁹ [assignment: *access control SFP*]

¹¹⁰ [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

¹¹¹ [assignment: *access control SFP*]

(d) *Certified Migration Key Blob with the security attributes payload type and migrationAuth*¹¹².

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

- (1) *The Entity owner of a certifiable migratable User key is allowed to create a Wrapped Key Blob for this migratable key by means of the command TPM_CMK_CreateKey, if it is authorized by use of the CMK Migration Approval Ticket and in case of delegated commands the restrictions for the migration of keys are fulfilled.*
- (2) *The Entity owner of a migratable User key authorized for use of the Migration key authorization ticket is allowed to create a Migration Key Blob for this migratable key by means of the command TPM_CreateMigrationBlob,*
- (3) *The Entity owner of a certifiable migratable User key authorized for use of the Migration key authorization ticket and the Restriction Ticket is allowed to create a Certified Migration Key Blob for this migratable key by means of the command TPM_CMK_CreateBlob,*
- (4) *The Entity owner of private part of the migration User key is allowed to migrate a Migration Key Blob and a Certified Migration Key Blob to a conversion key by means of the command TPM_MigrateKey,*
- (5) *The Entity owner of the private part of migration User key is allowed to convert a Migration Key Blob by means of the command TPM_ConvertMigrationBlob and a Certified Migration Key Blob by means of the command TPM_CMK_ConvertMigration if in case of delegated commands the restrictions for the migration of keys are fulfilled*¹¹³.

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

Application note 24: The security attributes *payload* and *migrationAuth* are defined in [6], sec. 4.2 and 10 (cf. to table 1 for details). Note the security attribute *restrictDelegate* defined

¹¹² [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*]

¹¹³ [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

for the TPM applies for entity owners using delegation (cf. to command TPM_CMK_SetRestrictions in [7], sec. 11.5, for details). The security attribute *migrationScheme* authorizes migration keys to be used by the entity owner (cf. TPM_AuthorizeMigrationKey in [7], sec. 11.3 for details).

FMT_MSA.1/MigK Management of security attributes

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

FMT_MSA.1.1/MigK The TSF shall enforce the *Key Migration SFP*¹¹⁴ to restrict the ability to *assign initial value*¹¹⁵ the security attributes *restrictDelegate*, *migrationScheme*, *migrationAuthorityApproval*¹¹⁶ to *TPM owner*¹¹⁷.

FMT_MTD.1/MigK Management of TSF data

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

FMT_MTD.1.1/MigK The TSF shall restrict the ability to *create*¹¹⁸ the *CMK Migration Approval Ticket*, *Migration Key Authorization Ticket*, *Restrict Ticket*¹¹⁹ to *TPM owner*¹²⁰.

Application note 25: Migration key authorization blob is the data that the TPM owner creates by the command TPM_AuthorizeMigrationKey (referred as outData with a TPM_MIGRATIONKEYAUTH structure in [7], sec. 11.3) and the key owner uses by for authorization of export the migratable key by TPM_CreateMigrationBlob (cf. [7], sec. 11.1).

6.1.5.3. Measurement and Reporting

¹¹⁴ [assignment: *access control SFP*, *information flow control SFP*]

¹¹⁵ [selection: *change_default*, *query*, *modify*, *delete*, [assignment: *other operations*]]

¹¹⁶ [assignment: *list of security attributes*]

¹¹⁷ [assignment: *the authorised identified roles*]

¹¹⁸ [selection: *change_default*, *query*, *modify*, *delete*, *clear*, [assignment: *other operations*]]

¹¹⁹ [assignment: *list of TSF data*]

¹²⁰ [assignment: *the authorised identified roles*]

FDP_ACC.1/M&R Subset access control

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

FDP_ACC.1.1/M&R The TSF shall enforce the *Measurement and Reporting SFP*¹²¹ on

- (1) *Subjects: SHA-1 session, user "World" and entity owner,*
- (2) *Objects: PCR, User key,*
- (3) *Operations: commands TPM_SHA1Start, TPM_SHA1Update, TPM_SHA1Complete, TPM_SHA1CompleteExtend TPM_PCR_Reset, TPM_Extend, TPM_PCRRead, TPM_Quote TPM_Quote2, TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END*¹²².

FDP_ACF.1/M&R Security attribute based access control

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

FDP_ACF.1.1/M&R The TSF shall enforce the *Measurement and Reporting SFP*¹²³ to objects based on the following:

- (1) *Subjects:*
 - (a) *SHA-1 session,*
 - (b) *user with the security attributes locality,*
 - (c) *entity owner of the signature key with the security attribute usageAuth,*
- (2) *Objects:*
 - (a) *PCR with the security attributes pcrReset, pcrResetLocal, pcrExtendLocal*
 - (b) *User key with the security attribute keyUsage*¹²⁴.

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

- (1) *The SHA-1 session is allowed to reset the digest of the SHA-1 session by command TPM_SHA1Start.*

¹²¹ [assignment: *access control SFP*]

¹²² [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

¹²³ [assignment: *access control SFP*]

¹²⁴ [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*]

- (2) *The SHA-1 session is allowed to calculate the new digest of the SHA-1 session as SHA-1 hash value of the digest of the SHA-1 session and the presented data by command TPM_SHA1Update.*
- (3) *The SHA-1 session is allowed (i) to finish the calculation of the digest of the SHA-1 session as SHA-1 hash value of the digest of the SHA-1 session and the presented data and (ii) to output the hash value by command TPM_SHA1Complete.*
- (4) *The SHA-1 session is allowed (i) to finish the calculation of the digest of the SHA-1 session as SHA-1 hash value of the digest of the SHA-1 session and the presented data and (ii) to extend the value of the indicated PCR by command TPM_SHA1CompleteExtend.*
- (5) *If the pcrReset is TRUE the command TPM_Startup is allowed to set a PCR to 0xFF...FF.*
- (6) *If the pcrReset is FALSE the command TPM_Startup is allowed to set a PCR to 0x00...00.*
- (7) *If the user presents the locality matching the security attribute pcrResetLocal of the selected PCR and the pcrReset of this PCR is TRUE than the command TPM_PCR_Reset is allowed to reset this PCR to 0x00...00 or 0xFF...FF, where the concrete value is defined in the platform specific specification of the TOE.*
- (8) *If the user presents the locality matching the security attribute pcrExtendLocal of the selected PCR the command TPM_SHA1CompleteExtend is allowed (i) to finish the calculation of the digest of the SHA-1 session as SHA-1 hash value of the digest of the SHA-1 session and the presented data and (ii) to extend the value of the selected PCR with the final digest of the SHA-1 session.*
- (9) *If the user presents the locality matching the security attribute pcrExtendLocal of the selected PCR the command TPM_Extend is allowed to extend the value of the selected PCR with the presented data.*
- (10) *The user "World" is allowed to read the PCR object with the command TPM_PCRRead.*
- (11) *The entity owner is allowed to quote the PCR indicated by the parameter targetPCR with the User key, which security attribute keyUsage equals to TPM_KEY_SIGNING, TPM_KEY_IDENTITY, or TPM_KEY_LEGACY, by means of the command TPM_Quote or TPM_Quote2.*

(12) The user "World" under locality 4 is allowed to execute the LPC commands TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END.

(13) [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects].¹²⁵

FDP_ACF.1.3/M&R 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/M&R The TSF shall explicitly deny access of subjects to objects based on the [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects].

Application note 26: The ST writer shall perform the assignment in the element FDP_ACF.1.1/M&R, clause (12), depending on the platform specific specification of the TPM. The assignment may be empty ("none additional rules"). The ST writer may Example given the TPM_HASH_END LPC command will extend the identified PCR of locality 4 in case of a PC client specific TPM. Note, the PCR object is a set of PCR selected by the command. The PCR is selected by the parameter pcrSelection in the command TPM_PCR_Reset, and by the parameter pcrNum in the command TPM_SHA1CompleteExtend and TPM_Extend. The security attributes pcrReset, pcrResetLocal and pcrExtendLocal are TPM permanent data (cf. [6], sec. 8.8, for details). The entity owner of the signature key is authorized to use the key if the AuthData is valid for the use of the key pointed to by the key handle, the key handle point to SHA-1 signature scheme and - in case of the command TPM_Quote - to the valid key usage. The PCR object is a set of PCR selected by the parameter targetPCR in the command TPM_Quote or TPM_Quote2. The command TPM_Quote or TPM_Quote2 calculate a digital signature of the selected PCR and a challenge externalData presented by the user. The command TPM_SHA1CompleteExtend extends the PCR depending on the locality attribute presented by the user.

FMT_MSA.3/M&R Static attribute initialisation

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*¹²⁶ to provide *restrictive*¹²⁷ default values for security attributes that are used to enforce the SFP.

¹²⁵ [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

FMT_MSA.3.2/M&R The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.

FCO_NRO.1/M&R Selective proof of origin

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

FCO_NRO.1.1/M&R The TSF shall be able to generate evidence of origin for transmitted *TPM_QUOTE_INFO* or *TPM_QUOTE_INFO2* structure¹²⁸ at the request of the *originator*¹²⁹.

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

- (1) *PCR values of the requested PCR indices in case of TPM_QUOTE_INFO,*
- (2) *PCR values of the requested PCR indices, and locality at release in case of TPM_QUOTE_INFO2*¹³⁰

of the originator of the information, and

- (1) *external data in the TPM_QUOTE_INFO,*
- (2) *external data in the TPM_QUOTE_INFO2*¹³¹

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*¹³² given *the attributes of the Attestation Identity Key Credential if an Attestation Identity Key is used*¹³³.

Application note 27: The entity owner may use the command *TPM_Quote* or *TPM_Quote2* with keys of keyUsage *TPM_KEY_IDENTITY*, *TPM_KEY_SIGNING*, or *TPM_KEY_LEGACY*. The TSF generates the digital signature with the private portion of the key identified by the key handle. If the used key is an Attestation Identity Key (i.e. keyUsage is *TPM_KEY_IDENTITY*) the Attestation Identity Key Credential will provide information about the TPM as the origin of the signed data. If the keyUsage is *TPM_KEY_SIGNING*, or *TPM_KEY_LEGACY* a credential of this key may or may not exist. If no credential for the key exist than the command *TPM_Quote* or *TPM_Quote2* may not provide prove of origin. The

¹²⁶ [assignment: *access control SFP, information flow control SFP*]

¹²⁷ [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

¹²⁸ [assignment: *list of information types*]

¹²⁹ [selection: *originator, recipient, [assignment: list of third parties]*]

¹³⁰ [assignment: *list of attributes*]

¹³¹ [assignment: *list of information fields*]

¹³² [selection: *originator, recipient, [assignment: list of third parties]*]

¹³³ [assignment: *limitations on the evidence of origin*]

signed data are identified by the command TPM_Quote as TPM_QUOTE_INFO or TPM_Quote2 as TPM_QUOTE_INFO2. The data structures TPM_QUOTE_INFO TPM_QUOTE_INFO2 are defined in [5], sec. 11.3 and 11.4. The external data is part of the signed data and provided with these commands (cf. to [7], ch. 16.3 and 16.5 for details). TPM_SS_RSASSAPKCS1v15_SHA1 [5] is an approved signature scheme using RSA as signature algorithm, the hash function SHA-1 and the padding scheme RSASSA according to PKCS#1, version 2.0, sec. 8.1.

6.1.5.4. Non-volatile Storage

FDP_ACC.1/NVS Subset access control

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

FDP_ACC.1.1/NVS The TSF shall enforce the NVS SFP¹³⁴ on

- (1) *Subjects: user "World", entity owner and TPM owner,*
- (2) *Objects: NV storage areas,*
- (3) *Operations: create, write, read¹³⁵.*

FDP_ACF.1/NVS Security attribute based access control

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

FDP_ACF.1.1/NVS The TSF shall enforce the NVS SFP¹³⁶ to objects based on the following:

- (1) *Subjects: user "World", entity owner and TPM owner with the security attributes physical presence, locality and current PCR values,*
- (2) *Objects: NV storage with the security attributes nvLocked, noOwnerNWwrite, pcrInfoRead, pcrInfoWrite, localityAtRelease, and permissions TPM_NV_PER_READ_STCLEAR, TPM_NV_PER_WRITE_STCLEAR TPM_NV_PER_AUTHWRITE, TPM_NV_PER_OWNERWRITE TPM_NV_PER_PPWRITE,*

¹³⁴ [assignment: access control SFP]

¹³⁵ [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

¹³⁶ [assignment: access control SFP]

*TPM_NV_PER_AUTHREAD, TPM_NV_PER_PPREAD,
TPM_NV_PER_OWNERREAD, TPM_MAX_NV_WRITE_NOOWNER*¹³⁷.

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

- (1) *The user "World" under physical presence is allowed to create NV storage by means of the command TPM_NV_DefineSpace if nvLocked is 0 and noOwnerNVWrite does not exceed TPM_MAX_NV_WRITE_NOOWNER.*
- (2) *The TPM owner is allowed to create a NV storage area by means of the command TPM_NV_DefineSpace.*
- (3) *The user "World" is allowed to write the NV storage area if nvLocked of the TPM_PERMANENT_FLAGS is FALSE and max NV writes without an owner is not exceeded.*
- (4) *The TPM owner is allowed to write an NV storage area by means of the command TPM_NV_WriteValue if*
 - (a) *TPM_NV_PER_OWNERWRITE is TRUE,*
 - (b) *the user satisfies the requirement for physical presence defined in TPM_NV_PER_PPWRITE,*
 - (c) *the locality of the user mach the localityAtRelease defined for the TPM_NV_DATA_AREA and*
 - (d) *if pcrInfoWrite defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoWrite.*
- (5) *The entity owner is allowed to write an NV storage area by means of the command TPM_NV_WriteValueAuth if*
 - (a) *TPM_NV_PER_AUTHWRITE is TRUE,*
 - (b) *the user match the requirement for physical presence defined in TPM_NV_PER_PPWRITE,*
 - (c) *the locality of the user matches the localityAtRelease defined for the TPM_NV_DATA_AREA and*
 - (d) *if pcrInfoWrite defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoWrite.*
- (6) *The TPM owner is allowed to read an NV storage area by means of the command TPM_NV_ReadValue if*

¹³⁷ [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]

- (a) *TPM_NV_PER_OWNERREAD is TRUE,*
 - (b) *the user match the requirement for physical presence defined in TPM_NV_PER_PPREAD,*
 - (c) *the locality of the user matches the localityAtRelease defined in the pcrInfoRead and*
 - (d) *if pcrInfoRead defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoRead.*
- (7) *The Entity owner is allowed to read an NV storage area by means of the command TPM_NV_ReadValueAuth if*
- (a) *TPM_NV_PER_AUTHREAD is TRUE,*
 - (b) *the user matches the requirement for physical presence defined in TPM_NV_PER_PPREAD,*
 - (c) *the locality of the user matches the localityAtRelease defined in the pcrInfoRead and*
 - (d) *if pcrInfoRead defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoRead¹³⁸.*

FDP_ACF.1.3/NVS 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/NVS The TSF shall explicitly deny access of subjects to objects based on the rules:

- (1) *If TPM_NV_PER_READ_STCLEAR is TRUE the NV storage area can not be read after read with a data size of 0 until successful write or TPM_Startup(ST_Clear).*
- (2) *If TPM_NV_PER_WRITE_STCLEAR is TRUE the NV storage area can not be written after write to the specified index with a data size of 0 until TPM_Startup(ST_Clear).*
- (3) *If TPM_NV_PER_WRITEDEFINE is TRUE the NV storage area can not be written after performing the TPM_NV_DefineSpace command and one successful write to the index with datasize of 0.*
- (4) *If TPM_NV_PER_GLOBALLOCK is TRUE the NV storage area can not be written after successful write to index 0 until TPM_Startup(ST_Clear)*

¹³⁸ [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

- (5) [assignment: additional rules, based on security attributes, that explicitly deny access of subjects to objects]¹³⁹.

Application note 28: The ST writer shall perform the missing operation in the elements FDP_ACF.1.3/NVS and FDP_ACF.1.4/NVS where the assignment “none” is allowed. The access conditions for the listed commands are described in the TPM specification [6] and [7], chapter 20. The security attributes of the TPM_NV_PER_OWNERWRITE and TPM_NV_PER_AUTHWRITE are mutual exclusive. The security attributes of the TPM_NV_PER_OWNERREAD and TPM_NV_PER_AUTHREAD are mutual exclusive.

FMT_MSA.3/NVS Static attribute initialisation

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

FMT_MSA.3.1/NVS The TSF shall enforce the *NVS SFP*¹⁴⁰ to provide *restrictive*¹⁴¹ default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/NVS The TSF shall allow the *TPM owner and user “World” under physical presence*¹⁴² to specify alternative initial values to override the default values when an object or information is created.

6.1.5.5. Counter

FDP_ACC.1/MC Subset access control

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

FDP_ACC.1.1/MC The TSF shall enforce the *Monotonic Counter SFP*¹⁴³ on
(1) *Subjects: TPM owner, Delegated entity, Entity owner of the monotonic counter object, user “World”,*
(2) *Objects: Monotonic counter,*
(3) *Operations: create, increment, read, release*¹⁴⁴.

¹³⁹ [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

¹⁴⁰ [assignment: access control SFP, information flow control SFP]

¹⁴¹ [selection, choose one of: restrictive, permissive, [assignment: other property]]

¹⁴² [assignment: the authorised identified roles]

¹⁴³ [assignment: access control SFP]

¹⁴⁴ [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

FDP_ACF.1/MC Security attribute based access control

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

FDP_ACF.1.1/MC The TSF shall enforce the *Monotonic Counter SFP*¹⁴⁵ to objects based on the following:

- (1) *Subjects: TPM owner, Delegated entity, Entity owner of the monotonic counter object, user "World",*
- (2) *Objects: Monotonic counter with security attribute countID*¹⁴⁶.

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

- (1) *The TPM owner and Delegated entity are allowed to create a Monotonic counter, OSAP and DSAP sessions are required for creation of the Monotonic counter.*
- (2) *The Entity owner of the monotonic counter object is allowed to increment the Monotonic counter if the countID is set in TPM_STCLEAR_DATA for the current boot cycle.*
- (3) *The user "World" is allowed to read the Monotonic counter value if he addresses the Monotonic counter object correctly with valid countID.*
- (4) *The Entity owner of the monotonic counter object is allowed to release the Monotonic counter.*
- (5) *The TPM owner is allowed to release the Monotonic counter.*¹⁴⁷

FDP_ACF.1.3/MC 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/MC The TSF shall explicitly deny access of subjects to objects based on the rule:

¹⁴⁵ [assignment: *access control SFP*]

¹⁴⁶ [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*]

¹⁴⁷ [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

- (1) *The TSF shall disallow the operation read or increment the monotonic counter if the countID is invalid*¹⁴⁸.

FMT_MSA.1/MC Management of security attributes

Hierarchical to: No other components.

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

FMT_MSA.1.1/MC The TSF shall enforce the *Monotonic Counter SFP*¹⁴⁹ to restrict the ability to

- (1) *modify*¹⁵⁰ the security attributes *countID*¹⁵¹ to the *Entity owner executing TPM_IncrementCounter*¹⁵².
- (2) *set to NULL*¹⁵³ the security attributes *countID*¹⁵⁴ to *TPM_Startup(ST_CLEAR)*¹⁵⁵,
- (3) *set to invalid value*¹⁵⁶ the security attributes *countID*¹⁵⁷ to
- (a) *Entity owner of the monotonic counter executing the command TPM_ReleaseCounter*
- (b) *TPM owner executing the command TPM_ReleaseCounterOwner*¹⁵⁸.

FMT_MSA.3/MC Static attribute initialisation

Hierarchical to: No other components.

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

¹⁴⁸ [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

¹⁴⁹ [assignment: *access control SFP, information flow control SFP*]

¹⁵⁰ [selection: *change_default, query, modify, delete, [assignment: other operations]*]

¹⁵¹ [assignment: *list of security attributes*]

¹⁵² [assignment: *the authorised identified roles*]

¹⁵³ [selection: *change_default, query, modify, delete, [assignment: other operations]*]

¹⁵⁴ [assignment: *list of security attributes*]

¹⁵⁵ [assignment: *the authorised identified roles*]

¹⁵⁶ [selection: *change_default, query, modify, delete, [assignment: other operations]*]

¹⁵⁷ [assignment: *list of security attributes*]

¹⁵⁸ [assignment: *the authorised identified roles*]

FMT_MSA.3.1/MC The TSF shall enforce the *Monotonic Counter SFP*¹⁵⁹ to provide *restrictive*¹⁶⁰ default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/MC The TSF shall allow *no role*¹⁶¹ to specify alternative initial values to override the default values when an object or information is created.

FPT_STM.1 Reliable time stamps

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

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps **as number Tick Count Value of ticks since start of the tick session to an accuracy of tickRate microseconds.**

Application note 29: The Tick Count Value provided by the TPM is not an actual universal time clock (UTC) value but is the number of timer ticks the TPM from the start of a timing session. It is the responsibility of the caller to associate the ticks to an actual UTC time. The tickRate is set during manufacturing of the TPM and platform (cf. [5], chapter 20). The Tick Count Value can be read by the user World using the command TPM_GetTicks.

FCO_NRO.1/STS Selective proof of origin

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

FCO_NRO.1.1/STS The TSF shall be able to generate evidence of origin for transmitted *TPM_SIGN_INFO structure*¹⁶² at the request of the *originator*¹⁶³.

FCO_NRO.1.2/STS The TSF shall be able to relate *the current tick count*¹⁶⁴ of the originator of the information, and *external data in the TPM_SIGN_INFO structure*¹⁶⁵ of the information to which the evidence applies.

FCO_NRO.1.3/STS The TSF shall provide a capability to verify the evidence of origin of information to *recipient*¹⁶⁶ given *the attributes of the Attestation Identity Key Credential if an Attestation Identity Key is used*¹⁶⁷

¹⁵⁹ [assignment: *access control SFP, information flow control SFP*]

¹⁶⁰ [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

¹⁶¹ [assignment: *the authorised identified roles*]

¹⁶² [assignment: *list of information types*]

¹⁶³ [selection: *originator, recipient, [assignment: list of third parties]*]

¹⁶⁴ [assignment: *list of attributes*]

¹⁶⁵ [assignment: *list of information fields*]

Application note 30: The command `TPM_TickStampBlob` signs the presented data `digestToStamp` in the `TPM_SIGN_INFO` structure after concatenation with the `TPM_CURRENT_TICKS` structure containing the current Tick Count Value using the signature key. The entity owner may use the command `TPM_TickStampBlob` with keys of keyUsage `TPM_KEY_IDENTITY`, `TPM_KEY_SIGNING`, or `TPM_KEY_LEGACY`. The TSF generates the digital signature with the private portion of the key identified by the key handle. If the used key is an Attestation Identity Key (i.e. keyUsage is `TPM_KEY_IDENTITY`) the Attestation Identity Key Credential will provide information about the TPM as the origin of the signed data. If the keyUsage is `TPM_KEY_SIGNING`, or `TPM_KEY_LEGACY` a credential of this key may or may not exist. If no credential for the key exists then the command `TPM_Quote` or `TPM_Quote2` may not provide prove of origin.

6.1.6. Data Import and Export

FDP_ACC.1/EID Subset access control

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

FDP_ACC.1.1/IED The TSF shall enforce the *Export and Import of Data SFP*¹⁶⁸ on
(1) Subjects: *TPM owner, Entity owner*;
(2) Objects: *Sealed Data, Context, Bound Blob*;
(3) Operations: *export, import, save, load, unbind*¹⁶⁹.

FDP_ACF.1/EID Security attribute based access control

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

FDP_ACF.1.1/EID The TSF shall enforce the *Export and Import of Data SFP*¹⁷⁰ to objects based on the following:
(1) Subjects: *TPM owner with security attribute locality, Entity owner with security attribute locality, user "World"*;
(2) Objects:
(a) *Sealed data with security attribute pcrInfo and tpmProof*,
(b) *Context with the security attribute resourceType and tpmProof*,

¹⁶⁶ [selection: *originator, recipient, [assignment: list of third parties]*]

¹⁶⁷ [assignment: *limitations on the evidence of origin*]

¹⁶⁸ [assignment: *access control SFP*]

¹⁶⁹ [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

¹⁷⁰ [assignment: *access control SFP*]

(c) *Bound Blob with the security attributes payload type*¹⁷¹.

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

- (1) *The Entity owner of the key to be used for export of sealed data is allowed to export Sealed Data if this export key has the security attribute TPM_KEY_STORAGE and is not migratable.*
- (2) *The Entity owner of the key to be used for import of sealed data is allowed to import Sealed Data if*
 - (a) *this import key has the security attribute TPM_KEY_STORAGE and is not migratable,*
 - (b) *the security attributes pcrInfo of sealed data blob shall match the values in the PCR indicated by pcrInfo,*
 - (c) *the security attributes tmpProof of sealed data blob shall match the values tmpProof in the TPM_PERMANENT_DATA of the TOE.*
- (3) *The user "World" is allowed to save Context if the resourceType is TPM_RT_KEY, TPM_RT_AUTH, TPM_RT_TRANS or TPM_RT_DAA_TPM.*
- (4) *The user "World" is allowed to load Context if*
 - (a) *the resourceType is TPM_RT_KEY, TPM_RT_AUTH, TPM_RT_TRANS or TPM_RT_DAA_TPM and*
 - (b) *the tmpProof used as secret for the HMAC of the context matches the tmpProof in TPM_PERMANENT_DATA.*
- (5) *The Entity owner of the private part of the bind key is allowed to unbind a Bound blob if the payload type is TPM_PT_BIND.*¹⁷²

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

¹⁷¹ [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*]

¹⁷² [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

Application note 31: The exported key blob contains security attributes as defined in the export format TPM_KEY (cf. [6], sec. 10.2) or TPM_KEY12 (cf. [6], sec. 10.3). The security attributes shall include authDataUsage, keyFlags, keyUsage and algorithmParms. The security attributes may include PCRInfo with the format TPM_PCR_INFO in case of the export format TPM_KEY and TPM_PCR_INFO_LONG in case of the export format TPM_KEY12.

FMT_MSA.3/EID Static attribute initialisation

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

FMT_MSA.3.1/EID The TSF shall enforce the *Export and Import of Data SFP*¹⁷³ to provide *restrictive*¹⁷⁴ default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/EID The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.

FDP_ETC.2 Export of user data with security attributes

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

FDP_ETC.2.1 The TSF shall enforce the *Key Management SFP, Key Migration SFP, Export and Import of Data SFP*¹⁷⁵ when exporting user data, controlled under the SFP(s), outside of the TOE.

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

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

¹⁷³ [assignment: *access control SFP, information flow control SFP*]

¹⁷⁴ [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

¹⁷⁵ [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

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

- (1) *User keys exported by means of the command TPM_CreateWrapKey shall be exported with the security attributes*
 - (a) *keyUsage,*
 - (b) *keyFlags,*
 - (c) *algorithmParms and*
 - (d) *PCRInfo with structure identified in KeyInfo if the key is bound to PCRs.*
- (2) *AIK keys shall be exported with the security attributes*
 - (a) *keyUsage,*
 - (b) *keyFlags,*
 - (c) *algorithmParms and*
 - (d) *PCRInfo with structure identified in idKeyParms.*
- (3) *Migration key blobs shall be exported with the security attributes*
 - (a) *keyUsage,*
 - (b) *keyFlags,*
 - (c) *algorithmParms and*
 - (d) *PCRInfo with structure identified in KeyInfo if the key is bound to PCRs.*
- (4) *Certified migration key blobs shall be exported with the security attributes*
 - (a) *keyUsage,*
 - (b) *keyFlags,*
 - (c) *algorithmParms and*
 - (d) *PCRInfo with structure TPM_PCR_INFO_LONG.*
- (5) *Sealed Data shall be exported with the security attributes pcrInfo and tpmProof.*
- (6) *Context shall be exported with the security attributes resource type and use the tpmProof as secret for the HMAC of the context¹⁷⁶.*

Application note 32: The structure of the security attribute PCRInfo exported with the User key depends on the used command.

FDP_ITC.2 Import of user data with security attributes

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

¹⁷⁶ [assignment: *additional exportation control rules*]

FPT_TDC.1 Inter-TSF basic TSF data consistency

- FDP_ITC.2.1 The TSF shall enforce the *Key Management SFP, Key Migration SFP, Export and Import of Data SFP*¹⁷⁷ when importing user data, controlled under the SFP, from outside of the TOE.
- FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.
- FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.
- FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.
- FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:
- (1) *User keys imported by means of the command TPM_LoadKey2 shall be imported with the security attributes contained in Wrapped key blob*¹⁷⁸.

FDP_UCT.1/Exp Basic data exchange confidentiality

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

FDP_UCT.1.1/Exp The TSF shall enforce the *Key Management SFP, Key Migration SFP, Export and Import of Data SFP*¹⁷⁹ to be able to *transmit*¹⁸⁰ user data

- (1) data together with the security attributes pcrInfo of an imported sealed data,**
- (2) migratable key of an imported Migration Key Blob or Certified Migration Key Blob,**
- (3) private portion of the key of an imported Wrapped Key Blob,**

¹⁷⁷ [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

¹⁷⁸ [assignment: *additional exportation control rules*]

¹⁷⁹ [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

¹⁸⁰ [selection: *transmit, receive*]

(4) data of the TPM_CONTEXT_SENSITIVE structure in the exported Context,

in a manner protected from unauthorised disclosure.

FDP_UCT.1/Imp Basic data exchange confidentiality

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

FDP_UCT.1.1/Imp The TSF shall enforce the *Key Management SFP, Key Migration SFP, Export and Import of Data SFP*¹⁸¹ **by providing the ability** to receive¹⁸² user data

- (1) data together with the security attributes TPM_PCR_INFO in a sealed data object,**
- (2) migratable key exported in a created or converted Migration Key Blob,**
- (3) migratable key exported in a created or converted Certified Migration Key Blob,**
- (4) private portion of the key exported in a Wrapped Key Blob,**
- (5) data of the TPM_CONTEXT_SENSITIVE structure in the loaded context,**
- (6) data of the wrapped command within a transport session**

in a manner protected from unauthorised disclosure.

Application note 33: The element FDP_UCT.1.1/Imp was refined by substituting “the TSF shall enforce ... to be able to” by “the TSF shall enforce... **by providing the ability** to” to emphasize that sender of user data shall ensure the confidentiality by encryption. The TSF shall be able to receive encrypted user data and to decrypt them when the user data are imported. Moreover the data protected by encryption are explicitly listed by refinements in FDP_UCT.1.1/Exp and FDP_UCT.1.1/Imp.

FDP_UIT.1/Data Data exchange integrity

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

¹⁸¹ [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

¹⁸² [selection: *transmit, receive*]

FTP_TRP.1 Trusted path]

FDP_UIT.1.1/Data The TSF shall enforce the *Key Management SFP, Key Migration SFP, Export and Import of Data SFP*¹⁸³ to be able to *transmit and receive*¹⁸⁴ user data in a manner protected from *modification, deletion and insertion*¹⁸⁵ errors.

FDP_UIT.1.2/Data The TSF shall be able to determine on receipt of user data

- (1) exported key,
- (2) migratable key and its security attributes in a created or converted Migration Key Blob,
- (3) migrated migratable key and its security attributes in a Wrapped Key,
- (4) certified migratable key and its security attributes in a created or converted Certified Migration Key Blob,
- (5) migrated Certified Migratable Key and its security attributes in a Wrapped Key Blob,
- (6) saved Context,

whether *modification, deletion and insertion*¹⁸⁶ has occurred.

FDP_UIT.1/Session Data exchange integrity

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

FDP_UIT.1.1/Session The TSF shall enforce the *TPM Mode Control SFP, Delegation SFP, Measurement and Reporting SFP, NVS SFP, Monotonic Counter SFP Key Management SFP, Key Migration SFP, Export and Import of Data SFP*¹⁸⁷ to be able to *transmit and receive*¹⁸⁸

- (1) command input,
- (2) return output data and

¹⁸³ [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

¹⁸⁴ [selection: *transmit, receive*]

¹⁸⁵ [selection: *modification, deletion, insertion, replay*]

¹⁸⁶ [selection: *modification, deletion, insertion, replay*]

¹⁸⁷ [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

¹⁸⁸ [selection: *transmit, receive*]

(3) ordinal, header information and data of the wrapped command in a transport session

in a manner protected from *modification, deletion, insertion and replay*¹⁸⁹ errors.

FDP_UIT.1.2/Session The TSF shall be able to determine on receipt of user data **command input**, whether *modification, deletion and insertion and replay*¹⁹⁰ has occurred.

Application note 34: The element FDP_UIT.1.1/Session was refined by substituting “user data” by “command input and return output data” to address the concrete type of user data. The subject “session” may run an Object-Independent Authorization Protocol (OIAP, [5], sec. 13.2.1), an Object-Specific Authorization Protocol (OSAP, [5], sec. 13.3), an Authorization-Data Insertion Protocol (ADIP, [5], sec. 13.5), an AuthData Change Protocol (ADCP, [5], sec. 13.6) or an Asymmetric Authorization Change Protocol (AACCP, [5], sec. 13.7). The TSF sends the field resAuth which is the HMAC of the output data including the rolling nonce the TPM returns to the caller (cf. to [5], sec. 13.1). The SFR FDP_UIT.1/Session addresses the integrity protection of the command response sent within the session. This integrity protection is a security feature of the session. The SFR FDP_UIT.1/Data addresses the integrity protection of data objects exported by the TOE which is a security feature of the TPM command.

Application note 35: The creation of sessions allows for the grouping of a set of commands into a session. Depending on the attributes specified for the session this may establish shared secrets, encryption keys, and session logs. The TOE decrypts and checks the integrity of the commands send to the TOE in the transport session.

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.
Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:
a) Start-up and shutdown of the audit functions;
b) All auditable events for the *not specified*¹⁹¹ level of audit; and
c) *Transport session*¹⁹².

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

¹⁸⁹ [selection: *modification, deletion, insertion, replay*]

¹⁹⁰ [selection: *modification, deletion, insertion, replay*]

¹⁹¹ [selection, choose one of: *minimum, basic, detailed, not specified*]

¹⁹² [assignment: *other specifically defined auditable events*]

- a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
- b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST,
- c) *Signed hash value of the TPM_TRANSPORT_LOG_IN structures of the received commands and TPM_TRANSPORT_LOG_OUT structures of the command responses*¹⁹³.

Application note 36:

FAU_GEN.1 describes the requirements for audit data generation of only one event: the transport session. The audit data is a signed hash value of the TPM_TRANSPORT_LOG_IN structures of the received commands and TPM_TRANSPORT_LOG_OUT structures of the command responses (cf. [6], chapter 13 for details). The transport logging data are generated by the TPM on demand of the entity establishing the transport session. The audit data generation starts up with the command TPM_EstablishTransport and shutdown with TPM_ReleaseTransportSigned.

FAU_GEN.1 is implemented by the transport session commands as follow:

- TPM_EstablishTransport:

The transport session audit is started for a transport session by setting the bit TPM_TRANSPORT_LOG in the transAttributes parameter of the command TPM_EstablishTransport. starting the audit function by calculating

$transDigest := SHA-1(TPM_TRANSPORT_LOG_OUT \text{ of the } TPM_EstablishTransport)$

- TPM_ExecuteTransport: collecting the TPM_TRANSPORT_LOG_IN structure of the received command and TPM_TRANSPORT_LOG_OUT structure of the command response and calculating

$transDigest := SHA-1(SHA-1(transDigest, TPM_TRANSPORT_LOG_IN), TPM_TRANSPORT_LOG_OUT)$

- TPM_ReleaseTransportSigned: creates the TPM_TRANSPORT_LOG_OUT structure of the command response, calculates

$transDigest := SHA-1(transDigest, TPM_TRANSPORT_LOG_OUT)$

and returns the signed transDigest as audit data.

The TPM_TRANSPORT_LOG_IN structure contains the ordinal of the received command and size and encrypted data of the command. The TPM_TRANSPORT_LOG_OUT structure contains the return code (i.e. information about success or failure of command execution), locality of the command and the current ticks, when command responses were sent.

¹⁹³ [assignment: *other audit relevant information*]

6.1.7. DAA

This section describes security functional requirements for implementation of the TPM part of the Direct Anonymous Attestation Protocol [5], chapter 33.

FDP_ACC.1/DAA Subset access control - DAA

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

FDP_ACC.1.1/DAA The TSF shall enforce the *DAA SFP*¹⁹⁴ on
(1) *Subjects: TPM owner*,
(2) *Objects: DAA_tpmSpecific*,
(3) *Operations: commands TPM_DAA_Join, TPM_DAA_Sign*¹⁹⁵.

FDP_ACF.1/DAA Security attribute based access control - DAA

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

FDP_ACF.1.1/DAA The TSF shall enforce the *DAA SFP*¹⁹⁶ to objects based on the following:
(1) *Subjects: TPM owner*,
(2) *Objects: DAA_tpmSpecific*¹⁹⁷.

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

(1) *The TPM owner is allowed to execute the commands TPM_DAA_Join and TPM_DAA_Sign*¹⁹⁸.

FDP_ACF.1.3/DAA The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*¹⁹⁹.

¹⁹⁴ [assignment: *access control SFP*]

¹⁹⁵ [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

¹⁹⁶ [assignment: *access control SFP*]

¹⁹⁷ [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*]

¹⁹⁸ [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

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

- (1) *The TSF shall disallow the TPM_DAA_Sign if the DAA_tpmSpecific is not generated by the same TOE²⁰⁰.*

Application note 37: The command TPM_DAA_Join generates DAA_issuerSettings and DAA_tpmSpecific as part of the TPM_STANY_DATA structure in the TPM and exports encrypted DAA_tpmSpecific, v0 and v1. The command TPM_DAA_Sign verifies that the DAA parameter DAA_tpmSpecific, v0 and v1 provided in encrypted form are in as being created by the same TPM and proves the attestation held by a TPM without revealing the attestation held by that TPM.

FMT_MSA.1/DAA Management of security attributes

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

FMT_MSA.1.1/DAA The TSF shall enforce the *DAA SFP²⁰¹* to restrict the ability to *modify²⁰²* the security attributes *DAA parameters²⁰³* to the *Entity owner²⁰⁴*.

FMT_MSA.3/DAA Static attribute initialisation

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

FMT_MSA.3.1/DAA The TSF shall enforce the *DAA SFP²⁰⁵* to provide *restrictive²⁰⁶* default values for security attributes that are used to enforce the SFP.

¹⁹⁹ [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

²⁰⁰ [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

²⁰¹ [assignment: *access control SFP, information flow control SFP*]

²⁰² [selection: *change_default, query, modify, delete, [assignment: other operations]*]

²⁰³ [assignment: *list of security attributes*]

²⁰⁴ [assignment: *the authorised identified roles*]

²⁰⁵ [assignment: *access control SFP, information flow control SFP*]

²⁰⁶ [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

FMT_MSA.3.2/DAA The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.

FPR_UNL.1 Unlinkability

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

FPR_UNL.1.1 The TSF shall ensure that *users*²⁰⁷ are unable to determine whether *Direct Anonymous Attestation with randomized base name of the verifier*²⁰⁸ is related as follows: *performed by the same identity*²⁰⁹.

Application note 38: The DAA issuer is a special entity in the TOE environment involved in the execution of the Direct Anonymous Attestation Protocol by the TPM owner as user of the TPM. If the verifier's base name is used in the DAA sign protocol, the verifier is able to link signatures generated by the same TPM.

6.1.8. TSF Protection

FPT_FLS.1 Failure with preservation of secure state

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

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: *failure of any crypto operations including RSA encryption, RSA decryption, SHA-1, RNG, RSA signature generation, HMAC generation; failure of any commands or internal operations, [assignment: list of additional types of failures in the TSF]*²¹⁰.

Application note 39: The ST writer shall perform the missing operation in the element FPT_FLS.1.1 according to the additional types of failures for which the TSF preserve a secure state. The assignment may be "none" if no additional types of failures are handled by the TSF.

²⁰⁷ [assignment: *set of users and/or subjects*]

²⁰⁸ [assignment: *list of operations*]

²⁰⁹ [selection: *were caused by the same user, are related as follows*[assignment: *list of relations*]]

²¹⁰ [assignment: *list of types of failures in the TSF*]

FPT_TST.1 TSF testing

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

FPT_TST.1.1 The TSF shall run a suite of self tests *at the request of an authorised user, at the condition: after each power-on and reset, prior to execution of the first call to a capability that uses those functions*²¹¹ to demonstrate the correct operation of the TSF operation of *the TSF*²¹².

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of *TSF data*²¹³.

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

Refinement:

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

- a. TPM_SHA1Start,
- b. TPM_SHA1Update,
- c. TPM_SHA1Complete,
- d. TPM_SHA1CompleteExtend,
- e. TPM_Extend,
- f. TPM_Startup,
- g. TPM_ContinueSelfTest.

Application note 40: The self-test capabilities are designed to enable the creation of a trusted platform with minimum latency due to TPM self-test. It might be possible to avoid wasting time, waiting for a TPM to do self-test, by designing a platform where TPM self-testing is done in parallel with other system functions, at a time when TPM capabilities are not required. The self-test is required at power-on and reset (cf. [5], ch. 9.2).

Tests will include a TPM automatically tests just those internal functions that are used by critical TPM capabilities. This permits the use of those critical TPM capabilities as soon as possible after start-up. Remaining TPM capabilities use additional internal functions that must be tested before the remaining TPM capabilities can execute. A test of the additional functions can be explicitly called. Alternatively, those functions will automatically be tested prior to execution of the first call to a capability that uses those functions. At any time, other self-test commands will explicitly cause the TPM to do a full self-test.

²¹¹ [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]*]

²¹² [selection: *[assignment: parts of TSF], the TSF*]

²¹³ [selection: *[assignment: parts of TSF], TSF data*]

TPM_SelfTestFull causes the TPM to do a full self-test. TPM_ContinueSelfTest causes the TPM to test the TPM internal functions that were not tested at start-up. TPM_ContinueSelfTest is unusual, in that it returns a result code to the caller before execution of the command and does not return a result code to the caller after execution of the command. If the functions used by a capability have not been tested, TPM_ContinueSelfTest is executed automatically after that capability is called and before it is executed. It is anticipated that the caller or TPM driver software is preprogrammed with knowledge of the time that the TPM will require to complete TPM_ContinueSelfTest. It is anticipated that a call to a TPM that is executing TPM_ContinueSelfTest would result in a “busy” indication.

The tests themselves only return a TPM_SUCCESS or TPM_FAILEDSELFTEST answer. TPM_GetTestResult must be used to discover why self-test failed. Upon the failure of a self-test the TPM goes into failure mode and does not allow most other operations to continue. These self-tests demonstrate the correct operation of the TSF.

(End of application note 40)

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]*²¹⁴ to the TSF²¹⁵ by responding automatically such that the SFRs are always enforced.

Application note 41: Physical tampering scenarios addressed in the element FPT_PHP.3.1 require physical access to the TPM. An interaction with the TOE can be done through the physical interfaces, which are realized using contacts, or through the chip surface. Physical manipulation refers to specific attacks where the TOE and its functional behavior are not only influenced but definite changes are made by applying mechanical, chemical and other methods. Physical probing pertains to “measurements” or active affecting or both using galvanic contacts or any type of charge interaction. Information leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. The concrete tampering scenarios depend on the TOE implementation and the attack potential against resistance is claimed in the ST (i.e. at least moderate attack potential as defined in this PP).

The ST writer shall perform the missing operation in the element FPT_PHP.3.1 by adding specific physical tampering scenarios for which resistance is claimed for the specific TOE. This assignment may be empty.

²¹⁴ [assignment: *physical tampering scenarios*]

²¹⁵ [assignment: *list of TSF devices/elements*]

6.2. Security Assurance Requirements for the TOE

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

Table 8: 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
	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.2 Testing: security enforcing modules
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.4 Methodical vulnerability analysis

6.3. Security Requirements Rationale

The table 9 provides an overview of the mapping between the security objective for the TOE and the functional security requirements.

Table 9: Security requirements rationale

	O.Anonymity	O.Context_Management	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Transport_Protection	O.DAA	O.Tamper_Resistance
FMT_SMR.1																	X					
FMT_SMF.1																X						
FMT_MSA.2																X						
FPT_TDC.1								X														
FCS_CKM.1			X												X							
FCS_RNG.1			X																			
FCS_CKM.4			X										X									
FCS_COP.1/SHA												X	X							X		
FCS_COP.1/HMAC		X	X		X		X	X	X											X		
FCS_COP.1/RSA_Sig													X		X							
FCS_COP.1/RSA_Enc			X		X			X												X		
FCS_COP.1/SymEnc		X	X		X			X												X		
FDP_ACC.1/Modes	X			X						X												
FDP_ACF.1/Modes	X			X						X												
FMT_MSA.1/Modes	X			X						X						X						
FMT_MSA.1/PhysP			X	X						X						X						
FMT_MTD.1/AuthData								X														
FMT_MTD.1/Deleg				X				X														
FIA_UID.1								X	X													
FIA_UAU.1								X														
FIA_UAU.4																			X	X		
FIA_UAU.5								X												X		
FIA_UAU.6								X														
FIA_AFL.1								X														

	O.Anonymity	O.Context_Management	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Transport_Protection	O.DAA	O.Tamper_Resistance
FMT_MTD.1/Lock							X															
FIA_USB.1			X				X			X						X						
FDP_RIP.1													X									
FDP_ACC.1/Deleg			X																			
FDP_ACF.1/Deleg			X							X												
FMT_MSA.1/DFT			X													X						
FMT_MSA.1/DT			X													X						
FMT_MSA.3/Deleg			X													X						
FDP_ACC.1/KeyMan		X												X								
FDP_ACF.1/KeyMan		X								X				X								
FMT_MSA.1/KeyMan		X												X	X							
FMT_MSA.1/KEvi		X														X						
FMT_MSA.3/KeyMan		X												X	X							
FDP_ACC.1/MigK		X																				
FDP_ACF.1/MigK		X																				
FMT_MSA.1/MigK		X														X						
FMT_MTD.1/MigK		X																				
FDP_ACC.1/M&R											X			X								
FDP_ACF.1/M&R										X	X			X								
FMT_MSA.3/M&R											X			X	X							
FCO_NRO.1/M&R										X		X		X								
FDP_ACC.1/NVS			X																			
FDP_ACF.1/NVS			X							X												
FMT_MSA.3/NVS			X													X						
FDP_ACC.1/MC			X																			
FDP_ACF.1/MC			X																			
FMT_MSA.1/MC			X													X						
FMT_MSA.3/MC			X													X						
FPT_STM.1												X								X		

	O.Anonymity	O.Context_Management	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Transport_Protection	O.DAA	O.Tamper_Resistance	
FCO_NRO.1/STS													X										
FDP_ACC.1/EID		X			X				X														
FDP_ACF.1/EID		X			X				X	X													
FMT_MSA.3/EID		X			X				X							X							
FDP_ETC.2		X	X		X						X												
FDP_ITC.2			X						X	X													
FDP_UCT.1/Exp		X	X		X																		
FDP_UCT.1/Imp		X	X						X												X		
FDP_UIT.1/Data		X	X		X				X														
FDP_UIT.1/Session																			X	X			
FAU_GEN.1																				X			
FDP_ACC.1/DAA																						X	
FDP_ACF.1/DAA																						X	
FMT_MSA.1/DAA																						X	
FMT_MSA.3/DAA																						X	
FPR_UNL.1																						X	
FPT_FLS.1						X												X					
FPT_TST.1							X											X					
FPT_PHP.3																							X

The table 9 demonstrates that each security objective for the TOE is covered by at least one security requirement.

6.3.1. Rationale for the Security Functional Requirements

O.Anonymity: The TOE must allow the user authenticated by operatorAuth and the user “World” under physical presence temporarily to deactivate the TPM and to hide the TPM attestation identity during a user session.

O.Anonymity is mapped to:

- FDP_ACC.1/Modes Subset access control, which defines the TPM Mode Control SFP, and FDP_ACF.1/Modes Security attribute based access control, which requires temporarily deactivated TOE to disallow execution of commands identified by empty cells in table 12, column Avail Deactivated, which includes any command using an AIK.
- FMT_MSA.1/Modes Management of security attributes requires that only the user authenticated by operatorAuth and user under physical presence are allowed to deactivate temporarily an enabled and active TPM.

O.Context_Management: The TOE must ensure a secure wrapping of a resource (except EK and SRK) 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.

O.Context_Management is mapped to:

- FDP_ACC.1/EID Subset access control, which defines the Export and Import of Data SFP, and the FDP_ACF.1/EID Security attribute based access control, which enforces access control for saving and loading Context blob including identification, restriction and checking the resource type of the data exported as context.
- FMT_MSA.3/EID Static attribute initialization requires restrictive default value for Context blob and FDP_ETC.2 Export of user data with security attributes requires to export Context blob with security attributes.
- FDP_UCT.1/Exp Basic data exchange confidentiality requires protection of confidentiality of the TPM_CONTEXT_SENSITIVE structure in the exported Context.
- FDP_UCT.1/Imp Basic data exchange confidentiality requires the ability to receive data of the TPM_CONTEXT_SENSITIVE structure in the loaded context in a manner protected from unauthorised disclosure.
- FDP_UIT.1/Data Data exchange integrity requires the TSF to save Context blob in a manner protecting the integrity and to monitor the integrity of received Context blob.
- FCS_COP.1/HMAC and FCS_COP.1/SymEnc provide the cryptographic function for protection of confidentiality and integrity of the Context blob.

O.Crypto_Key_Man: The TOE must manage cryptographic keys in a secure manner including generation of cryptographic keys using the TOE random number generator as source of randomness.

O.Crypto_Key_Man is mapped to:

- FDP_ACC.1/KeyMan Subset access control, which defines the Key Management SFP, and the FDP_ACF.1/KeyMan Security attribute based access control, which enforces access control for key creation, reading, usage, export, import and deletion of key and activation of AIK except migration (cf. FDP_ACC.1/MigK and FDP_ACF.1/MigK).
- FMT_MSA.1/KeyMan Management of security attributes and FMT_MSA.1/KEvi Management of security attributes require secure management of security attributes of keys.

- FMT_MSA.1/PhysP Management of security attributes limits the ability to assert and to manage of the security attribute physical presence used to control deletion of the SRK.
- FMT_MSA.3/KeyMan Static attribute initialisation requires restrictive default values of security attributes of keys.
- FDP_ACC.1/MigK Subset access control, which defines the Key Migration SFP, and the FDP_ACF.1/MigK Security attribute based access control, which enforces access control for key migration and certified key migration.
- FMT_MSA.1/MigK Management of security attributes requires secure management of security attributes of keys for key migration and certified key migration.
- FMT_MTD.1/MigK restricts the ability to create TSF data CMK Migration Approval Ticket, Migration Key Authorization Ticket, Restriction Ticket to the TPM owner.
- FDP_ETC.2 Export of user data with security attributes and FDP_ITC.2 Import of user data with security attributes ensure that keys are exported and imported together with their security attributes.
- FDP_UCT.1/Exp Basic data exchange confidentiality requires protection of confidentiality of the exported key.
- FDP_UCT.1/Imp Basic data exchange confidentiality requires the ability to import key in a manner protected from unauthorized disclosure.
- FDP_UIT.1/Data Data exchange integrity requires the TSF to export key in a manner protecting the integrity and to monitor the integrity of received key blob.
- FCS_CKM.1 Cryptographic key generation requires generation of RSA keys according to standards.
- FCS_RNG.1 Random number generation requires the TOE to provide a random umber generator, which may be used for generation of keys and signatures.
- FCS_CKM.4 Cryptographic key destruction requires destruction of keys according to standards.
- FCS_COP.1/HMAC, FCS_COP.1/RSA_Enc and FCS_COP.1/SymEnc provide the cryptographic function for protection of confidentiality and integrity of the key in key blobs.

O.DAC: The TOE must control and restrict user access to the TOE protected capabilities and shielded location 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.DAC is mapped to:

- FDP_ACC.1/Modes Subset access control, which defines the TPM Mode Control SFP, and FDP_ACF.1/Modes Security attribute based access control, which requires the TSF to disallow execution of commands identified not being available depending on the TOE mode identified by empty cells in table 12.
- FMT_MSA.1/Modes Management of security attributes defines the initial state of TOE mode and requires that only authorized roles are allowed to modify the TPM mode.

- FMT_MSA.1/PhysP Management of security attributes limits the ability to assert and to manage the security attribute physical presence used to enforce the Delegation SFP, and the NVS SFP implementing O.DAC.
- FIA_USB.1 User-subject binding associate the user security attributes with subjects acting on the behalf of that user for access control.
- FDP_ACC.1/Deleg Subset access control, which defines the Delegation SFP, and FDP_ACF.1/Deleg Security attribute based access control, which requires the TSF to provide a delegation mechanism to allow object owner to manage the access rights for their data objects using the principle of least privilege.
- FMT_MSA.1/DFT Management of security attributes and FMT_MSA.1/DT Management of security attributes describes the management of delegation family tables and delegation tables for access control for delegated commands.
- FMT_MSA.3/Deleg Static attribute initialization requires permissive default values for security attributes that are used to enforce the Delegation SFP.
- FMT_MTD.1/Deleg Management of TSF data describes requirements for the management of TSF data for delegation.
- FDP_ACC.1/NVS Subset access control, which defines the NVS SFP, and FDP_ACF.1/NVS Security attribute based access control, which requires controlling access to NV storage behaving as shielded locations.
- FMT_MSA.3/NVS Static attribute initialisation requires permissive default values for security attributes of NVS that are used to enforce the NVS SFP.
- FDP_ACC.1/MC Subset access control, which defines the *Monotonic Counter SFP*, and the FDP_ACF.1/MC Security attribute based access control, which enforces access control for creation, increment, reading and releasing a monotonic counter.
- FMT_MSA.1/MC Management of security attributes requires secure management of security attributes of monotonic counters.
- FMT_MSA.3/MC Static attribute initialisation requires restrictive default values of security attributes of a monotonic counter.

Note that specific requirements for key management as a protected capability is addressed by O.Crypto_Key_Man.

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.

O.Export is mapped to:

- FDP_ACC.1/EID Subset access control, which defines the Export and Import of Data SFP, and the FDP_ACF.1/EID Security attribute based access control, which enforces access control for export Sealed data and saving Context.

- FMT_MSA.3/EID Static attribute initialisation requires restrictive default values of security attributes of data blobs.
- FDP_ETC.2 Export of user data with security attributes ensures that keys are exported and imported together with their security attributes.
- FDP_UCT.1/Exp Basic data exchange confidentiality requires protection of confidentiality of the exported Sealed data and Context.
- FDP_UIT.1/Data Data exchange integrity requires the TSF to export key in a manner protecting the integrity and to monitor the integrity of received Sealed data and Context.
- FCS_COP.1/HMAC, FCS_COP.1/RSA_Enc and FCS_COP.1/SymEnc provide the cryptographic function for protection of confidentiality and integrity of Sealed data and Context.

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

O.Fail_Secure is mapped to FPT_FLS.1 Fail secure, which directly addresses the secure failure mode of the TOE.

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

O.General_Integ_Checks is mapped to FPT_TST.1 TSF testing, which requires the TSF to provide the capability to verify the integrity of TSF data and TSF executable code.

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

O.I&A is mapped to:

- FCS_COP.1/HMAC Cryptographic operation implements a part of the authentication mechanism for users.
- FMT_MTD.1/AuthData Management of TSF data restrict the ability to modify and to create authentication data for TPM Owner and Entity Owner. The authentication data operatorAuth may be created and modified by user under physical presence.
- FMT_MTD.1/Deleg Management of TSF data restricts the ability to create and modify the authentication data of a delegation blob to TPM Owner and authorized users according to specified rules.
- FIA_UID.1 Timing of identification ensures identification of users except identified commands and objects where entity owner has given the "World" access to.
- FIA_UAU.1 Timing of authentication ensures authentication of users except identified commands and objects where entity owner has given the "World" access to.
- FIA_UAU.5 Multiple authentication mechanisms requires specific authentication mechanisms and FIA_UAU.6 Re-authentication defines the condition for re-authentication of the user.

- FIA_AFL.1 Authentication failure handling requires detection and reaction when unsuccessful authentication attempts occur related to authentication attempts for the same user (cf. TPM dictionary attack mitigation mechanism).
- FMT_MTD.1/Lock Management of TSF data describe the reset of the Action Flag of TPM dictionary attack mitigation mechanism addressed by FIA_AFL.1.
- FIA_USB.1 User-subject binding associate the security attributes of the user with subjects acting on the behalf of that user.
- FCS_COP.1/HMAC Cryptographic operation requires implementation of HMAC which is used for authentication (authorization) of users by means of AuthData.

Note that the SFR FIA_UAU.4 prevents reuse of authentication verification data to fulfill O.I&A and O.Single_Auth.

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 authorized 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 (except the verification of the integrity of the data within a sealed data blob).

O.Import is mapped to:

- FDP_ACC.1/EID Subset access control, which defines the Export and Import of Data SFP, and the FDP_ACF.1/EID Security attribute based access control, which enforces access control for import Sealed data, loading Context and unbind Blob.
- FMT_MSA.3/EID Static attribute initialisation requires restrictive default values of security attributes of data blobs.
- FDP_ITC.2 Import of user data with security attributes ensures that keys are imported and imported together with their security attributes.
- FPT_TDC.1 Inter-TSF basic TSF data consistency, which ensures consistency of TSF data if imported.
- FDP_UCT.1/Imp Basic data exchange confidentiality requires protection of confidentiality of the imported Sealed data, Context and Bound Blob.
- FDP_UIT.1/Data Data exchange integrity requires the TSF to export key in a manner protecting the integrity and to monitor the integrity of received Sealed data, Context and Bound Blob.
- FCS_COP.1/HMAC, FCS_COP.1/RSA_Enc and FCS_COP.1/SymEnc provide the cryptographic function for protection of confidentiality and integrity of Sealed data, Context and Bound Blob.

Note O.Transport_Protection deals with transport sessions as special form of the communication and therefore addresses special aspects of import data in commands.

O.Limit_Actions_Auth: The TOE must restrict the actions a user may perform before the TOE verifies the identity of the user. This includes requirements for physical presence of the user.

O.Limit_Actions_Auth is mapped to:

- FDP_ACC.1/Modes Subset access control, which defines the TPM Mode Control SFP, and FDP_ACF.1/Modes Security attribute based access control, which requires the TSF to disallow execution of commands identified as unavailable for the unowned TOE mode by empty cells in table 12. This addresses specific action under physical presence of the user.
- FMT_MSA.1/Modes Management of security attributes defines the initial state of TOE mode and requires that only authorized roles are allowed to modify the TPM mode.
- FMT_MSA.1/PhysP Management of security attributes limits the ability to assert and to manage the security attribute physical presence used to enforce the TPM Mode Control SFP implementing O.Limit_Actions_Auth.

FIA_UID.1 Timing of identification lists the action allowed without identification of the user.

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

O.Locality is mapped to:

- FIA_USB.1 User-subject binding requires the TSF to associate the locality of the user with subjects acting on the behalf of that user.
- The SFR for security attribute based access control FDP_ACF.1/Deleg, FDP_ACF.1/KeyMan, FDP_ACF.1/M&R, FDP_ACF.1/NVS and FDP_ACF.1/EID enforce access control rules based on locality of the user.
- FDP_ETC.2 Export of user data with security attributes requires exporting and FDP_ITC.2 Import of user data with security attributes requires importing sealed data with security attribute locality and keys with security attribute locality if the export format TPM_KEY12 is used.
- FCO_NRO.1/M&R Selective proof of origin requires the TSF be able to generate evidence of origin for TPM_QUOTE_INFO2 structure of the PCR including locality at release.

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

O.Record_Measurement is mapped to:

- FCS_COP.1/SHA Cryptographic operation implements the hash function SHA-1 for calculating hash values.
- FDP_ACC.1/M&R Subset access control, which defines the Measurement and Reporting SFP and FDP_ACF.1/M&R Security attribute based access control, which requires the TSF to control the recording of the measurement results in the PCR.

- FMT_MSA.3/M&R Static attribute initialisation requires restrictive default values of security attributes of the PCR.

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.

O.MessageNR is mapped to:

- FCS_COP.1/SHA Cryptographic operation requires the TSF to implement SHA-1 and FCS_COP.1/RSA_Sig Cryptographic operation requires the TSF to implement RSA signature algorithm for signing PCR value and external data.
- FCO_NRO.1/M&R Selective proof of origin requires the TSF to be able to generate evidence of origin for TPM_QUOTE_INFO or TPM_QUOTE_INFO2 structure of the PCR at the request of the originator.
- FCO_NRO.1/STS Selective proof of origin requires the TSF to be able to generate evidence of origin for TPM_SIGN_INFO structure of the external data which may include the current tick count provided according to FPT_STM.1 Reliable time stamps.
- FPT_STM.1 Reliable time stamps requires the TSF to provide reliable time stamps as number Tick Count Value of ticks since start of the tick session to an accuracy of tickRate microseconds.

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.No_Residual_Info is directly mapped to

- FDP_RIP.1 Subset residual information protection, which requires the TSF to ensure that if de-allocation of a resources are performed on any object the information in those objects is irretrievably removed.
- FCS_CKM.4 Cryptographic key destruction, which requires destruction of keys by methods that comply with standards.

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

O.Reporting is mapped to:

- FCS_CKM.1 Cryptographic key generation requires the TSF to generate RSA keys, which includes EK and AIK.
- FCS_COP.1/RSA_sig Cryptographic operation requires the TSF to generate digital signatures, which are used for reporting.
- FDP_ACC.1/KeyMan Subset access control, which defines the Key Management SFP, and the FDP_ACF.1/KeyMan Security attribute based access control, which enforces access control for (i) creation, reading, and usage of EK and (ii) creation, export, and activation of AIK.

- FMT_MSA.1/KeyMan Management of security attributes requires secure management of security attributes of EK and AIK to enforce the Key Management SFP.
- FMT_MSA.3/KeyMan Static attribute initialisation requires restrictive default values of security attributes of keys including EK and AIK.
- FDP_ACC.1/M&R Subset access control, which defines the Measurement and Reporting SFP and FDP_ACF.1/M&Rs Security attribute based access control, which requires the TSF to control the PCR reset and the measurement in the PCR for reporting.
- FMT_MSA.3/M&R Static attribute initialisation requires restrictive default values of security attributes to enforce the Measurement and Reporting SFP.
- FCO_NRO.1/M&R Selective proof of origin requires the TSF be able to create digital signatures for TPM_QUOTE_INFO or TPM_QUOTE_INFO2 structure of the PCR with AIK.

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

O.Security_Attr_Mgt is mapped to:

- FMT_SMF.1 Specification of Management Functions including the management functions for security attributes.
- The iterations FMT_MSA.1/Modes, FMT_MSA.1/PhysP, FMT_MSA.1/DFT, FMT_MSA.1/DT, FMT_MSA.1/KeyMan, FMT_MSA.1/KEvi, FMT_MSA.1/MigK, and FMT_MSA.1/MC address management of security attributes by only authorized users for the TPM Mode Control SFP, Delegation SFP, Key Management SFP, Key Migration SFP and Monotonic Counter SFP.
- FMT_MSA.2 Secure security attributes, which ensures that only secure values are accepted for security attributes.
- The iterations FMT_MSA.3/Deleg, FMT_MSA.3/KeyMan, FMT_MSA.3/M&R, FMT_MSA.3/NVS, FMT_MSA.3/MC, and FMT_MSA.3/EID address initialization of security attributes by only authorized users for the Delegation SFP, Key Management SFP, Measurement and Reporting SFP, NVS SFP Monotonic Counter SFP and Export and Import of Data SFP.

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

O.Security_Roles is mapped to:

- FMT_SMR.1 Security roles requires to maintain the security roles TPM owner, Entity owner, Delegated entity, Entity user, user using operatorAuth and "World".

- FIA_USB.1 User-subject binding requires the TSF to associate the authData, authorization handle, authorization associated with the delegation blobs, locality of the user, physical presence with subjects acting on the behalf of that user.

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 an secure state in case of detected errors.

O.Self_Test is mapped to

- FPT_TST.1 TSF self-testing, which requires the TSF to run a suite of tests to demonstrate the correct operation of the TSF.
- FPT_FLS.1 Failure with preservation of secure state requires the TSF preserving a secure state when failures occur.

O.Single_Auth: The TOE must provide a single use authentication mechanism and require re-authentication to prevent “replay” and “man-in-the-middle” attacks.

O.Single_Auth is mapped to

- FIA_UAU.4 Single-use authentication requires the TSF to prevent successful reuse of authentication data.
- FIA_UAU.6 Re-authentication requires the TSF to re-authenticate the user under the condition that the user sent a command that requires authentication within a session.
- FDP_UIT.1/Session Data exchange integrity requires the TSF to link the authentication data for authorization a command with the command data and therefore to detect replay.

O.Transport_Protection: The TOE must provide the confidentiality of the payload of the commands within a transport session and the integrity of the transport log of commands.

O.Transport_Protection is mapped to:

- FCS_COP.1/HMAC, FCS_COP.1/RSA_Enc, FCS_COP.1/SHA, and FCS_COP.1/SymEnc require the TSF to implement cryptographic algorithms used for transport sessions to ensure the confidentiality of the data.
- FIA_UAU.4 Single-use authentication and FIA_UAU.5 Multiple authentication mechanisms protect the transport session by authenticating the sender of each command within the transport session by a shared secret and rolling nonces.
- FDP_UCT.1/Imp Basic data exchange confidentiality requires the TSF to receive data of the wrapped command within a transport session in a manner protected from unauthorised disclosure.
- FDP_UIT.1/Session Data exchange integrity requires the TSF to receive ordinal, header information and data of the wrapped command in a transport session in a manner protected from modification, deletion, insertion and replay errors.

- FAU_GEN.1 Audit data generation requires the TSF to generate transport session log of commands.
- FPT_STM.1 Reliable time stamps provides the as number Tick Count Value of ticks since start of the tick session to an accuracy of tickRate microseconds to be included in the audit data.

O.DAA: 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 Direct Anonymous Attestation Protocol.

O.DAA is implemented by

- FDP_ACC.1/DAA Subset access control – DAA and FDP_ACF.1/DAA Security attribute based access control – DAA requiring access control to the commands TPM_DAA_Join and TPM_DAA_Sign implementing the Direct Anonymous Attestation Protocol.
- FMT_MSA.1/DAA Management of security attributes restrict the ability to modify the security attributes DAA parameters to the Entity owner.
- FMT_MSA.3/DAA Static attribute initialization requires providing restrictive default values for security attributes that are used to enforce the DAA SFP.
- FPR_UNL.1 Unlinkability requires that users are unable to determine whether Direct Anonymous Attestation was performed by the same TPM, if the randomized base name is used in the DAA sign protocol.

O.Tamper_Resistance: The TOE must resist physical tampering of the TSF by hostile users.

O.Tamper_Resistance is implemented by

- FPT_PHP.3 requires the TPM to be resistant against identified physical tampering by an hostile user against the TSF by responding automatically such that the SFRs are always enforced.

6.3.2. Rationale for the Security Assurance Requirements

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 8.

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 manufacture 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 fulfils 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.

6.3.3. SFR Dependency Rationale

Table 10: SFR Dependency rationale

SFR	Dependency	Rationale
FMT_SMR.1	FIA_UID.1 Timing of identification	Fulfilled
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
FPT_TDC.1	No dependencies	n. a.
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]	Fulfilled

SFR	Dependency	Rationale
	FCS_CKM.4 Cryptographic key destruction	
FCS_RNG.1	No dependencies	n. a.
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 and FDP_ITC.2
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	Fulfilled, see rationale 1.
FCS_COP.1/HMAC	[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, see rationale 1.
FCS_COP.1/RSA_Sig	[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, FDP_ITC.2 and FCS_CKM.4.
FCS_COP.1/RSA_Enc	[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, FDP_ITC.2 and FCS_CKM.4.
FCS_COP.1/SymEnc	[FDP_ITC.1 Import of user	Fulfilled by FDP_ITC.2 and

SFR	Dependency	Rationale
	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_CKM.4.
FDP_ACC.1/Modes	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/Modes
FDP_ACF.1/Modes	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1/Modes fulfilled. For FMT_MSA.3 see Rationale 2.
FMT_MSA.1/Modes	[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/Modes
FMT_MSA.1/PhysP	[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/Key_Man
FMT_MTD.1/AuthData	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled
FMT_MTD.1/Deleg	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled
FIA_UID.1	No dependencies	n. a.
FIA_UAU.1	FIA_UID.1 Timing of identification	Fulfilled
FIA_UAU.4	No dependencies	n. a.
FIA_UAU.5	No dependencies	n. a.
FIA_UAU.6	No dependencies	n. a.
FIA_AFL.1	FIA_UAU.1 User authentication by TSF	Fulfilled
FMT_MTD.1/Lock	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled
FIA_USB.1	FIA_ATD.1 User attribute definition	See rationale 7

SFR	Dependency	Rationale
FDP_RIP.1	No dependencies	n. a.
FDP_ACC.1/Deleg	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/Deleg
FDP_ACF.1/Deleg	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/Deleg and FMT_MSA.3/Deleg
FMT_MSA.1/DFT	[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/Deleg
FMT_MSA.1/DT	[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/Deleg
FMT_MSA.3/Deleg	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/DFT, FMT_MSA.1/DT, FMT_SMR.1
FDP_ACC.1/KeyMan	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/KeyMan
FDP_ACF.1/KeyMan	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/KeyMan and FMT_MSA.3/KeyMan
FMT_MSA.1/KeyMan	[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/KeyMan, FMT_SMR.1 and FMT_SMF.1
FMT_MSA.1/KEvi	[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/KeyMan, FMT_SMR.1 and FMT_SMF.1
FMT_MSA.3/KeyMan	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/KeyMan and FMT_MSA.1/KEvi
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	Fulfilled by FDP_ACC.1/M&R

SFR	Dependency	Rationale
	control FMT_MSA.3 Static attribute initialisation	and FMT_MSA.3/M&R
FMT_MSA.3/M&R	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_SMR.1 and FMT_MSA.1/KeyMan for keys, see Rationale 3 for security attributes of the PCR
FCO_NRO.1/M&R	FIA_UID.1 Timing of identification	Fulfilled
FDP_ACC.1/NVS	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/NVS
FDP_ACF.1/NVS	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/NVS and FMT_MSA.3/NVS
FMT_MSA.3/NVS	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_SMR.1, see Rationale 4 for security attributes
FDP_ACC.1/MC	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/MC
FDP_ACF.1/MC	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/MC and FMT_MSA.3/MC
FMT_MSA.1/MC	[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/MC, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/MC	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/MC and FMT_SMR.1
FPT_STM.1	No dependencies	n. a.
FCO_NRO.1/STS	FIA_UID.1 Timing of identification	Fulfilled
FDP_ACC.1/MigK	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/MigK
FDP_ACF.1/MigK	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/MigK, see Rationale 5 for security attributes
FMT_MSA.1/MigK	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles	Fulfilled by FDP_ACF.1/MigK, FMT_SMR.1, FMT_SMF.1

SFR	Dependency	Rationale
	FMT_SMF.1 Specification of Management Functions	
FMT_MTD.1/MigK	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled
FDP_ACC.1/EID	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/EID
FDP_ACF.1/EID	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/EID and FMT_MSA.3/EID
FMT_MSA.3/EID	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_SMR.1, see Rationale 6 for security attributes
FDP_ETC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled
FDP_ITC.2	[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_ACC.1 and FPT_TDC.1 are fulfilled. See rationale 8.
FDP_UCT.1/Exp	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_ACC.1 is fulfilled. See rationale 8.
FDP_UCT.1/Imp	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_ACC.1 is fulfilled.. See rationale 8.
FDP_UIT.1/Data	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_ACC.1 is fulfilled. See rationale 8.
FDP_UIT.1/Session	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted	FDP_ACC.1 is fulfilled. See rationale 8.

SFR	Dependency	Rationale
	channel, or FTP_TRP.1 Trusted path]	
FAU_GEN.1	FPT_STM.1 Reliable time stamps	Fulfilled
FDP_ACC.1/DAA	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/DAA
FDP_ACF.1/DAA	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/DAA and FMT_MSA.3/DAA
FMT_MSA.1/DAA	[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/DAA, FMT_SMR.1 and FMT_SMF.1
FMT_MSA.3/DAA	FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/DAA and FMT_SMR.1
FPR_UNL.1	No dependencies	n. a.
FPT_FLS.1	No dependencies	n. a.
FPT_TST.1	No dependencies.	n. a.
FPT_PHP.3	No dependencies	n. a.

Rationale 1: The cryptographic algorithms SHA-1 and HMAC are applied to internal and external data. SHA-1 does not use any key. HMAC uses TSF data like authorization data instead of the key. Therefore neither import of keys, key generation nor key destruction are necessary.

Rationale 2: The rule (1) of the refinement of FMT_MSA.1.1/Modes requires the TPM being disabled, inactive and unowned when created. This requirement fits to the intention but is more restrictive as a default value of the security attribute as addressed by FMT_MSA.3.

Rationale 3: The security attributes pcrReset, pcrResetLocal, pcrExtendLocal of the PCR are values that are set during the manufacturing process of the TPM and platform and are not field settable or changeable values (cf. [6], sec. 8.7). Thus no management is available for these security attributes.

Rationale 4: The security attributes of the NVS can be set only by creating the NVS space but can not be changed after creation (cf. command TPM_NV_DefineSpace [7], sec. 20.1). The static attribute initialization is addressed by FMT_MSA.3/NVS.

Rationale 5: The security attribute *payload type* is defined for each object by execution of the command creating the object. The definition of the *migrationAuth* is described by management of TSF data, cf. SFR FMT_MTD.1/MigK.

Rationale 6: The security attributes of the exported data addressed in FDP_ACF.1/EID are set by static attribute initialisation (cf. FMT_MSA.3/EID) and can not be changed after creation of the objects.

Rationale 7: The user uses the TOE in specific roles listed in FMT_SMR.1 without distinguishing individual users. The authorization binds authentication and roles of the user for each session or command. Therefore FIA_ATD.1 is not necessary.

Rationale 8: The SFR FDP_UCT.1/Exp, FDP_UCT.1/Imp and FDP_UIT.1/Data ensure secure communication for the data objects. The SFR FDP_UIT.1/Session ensures integrity protected communication for each command. Therefore no trusted path or trusted channel is required here.

7. Annex

7.1. Ordinal table

The following table 12 is an extract of the ordinal table in [6], chapter 17, and the command descriptions in [7]. The column RQU is used in the definition of the SFR FIA_UID.1 which lists the command executable by users without authentication. The columns No owner, Avail Deactivated and Avail Disabled define the commands available in the TPM states. This table contains all rows except the rows of undefined ordinals.

Table 11: Description of the columns in table 12.

Column	Column Values	Comments and valid column entries
Optional	x	Is the command optional (and therefore not subject of this protection profile)
AUTH2	x	Does the command support two authorization entires, normally two keys
AUTH1	x	Does the commands support an single authorization session
RQU	x	Does the command execute without any authorization
No Owner	x	Is the command executable when no owner is present
Avail Deactivated	x, A	Ordinal will execute when deactivated A = Authorization means that command will only work if the underlying NV store does not require authorization
Avail Disabled	x, A	Ordinal will execute when disabled A = Authorization means that command will only work if the underlying NV store does not require authorization The TPM MUST return TPM_DISABLED for all commands other than those marked as avaiable
Physical presence	P, O, T	P = the command requires physical presence O = the command requires physical presence or operatorAuth authentication T = the command requires physical presence or TPM owner authentication T* = the NV space maybe configured to require physical presence additional to TPM owner authentication A* = the NV space maybe configured to require physical presence in addition to other entity owner authentication

Column	Column Values	Comments and valid column entries
PCR Use Enforced	x	Does the command enforce PCR restrictions when executed
Audit	x, N	Is the default for auditing enabled N = Never the ordinal is never audited
Duration	S, M, L	What is the expected duration of the command, S = Short implies no asymmetric cryptography M = Medium implies an asymmetric operation L = Long implies asymmetric key generation
1.2 Changes	N, D, X, C	N = New for 1.2 X = Deleted in 1.2 D = Deprecated in 1.2 C = Changed in 1.2
FIPS changes	x	Ordinal has change to satisfy FIPS 140 requirements

Table 12: TPM command ordinals and security features of the respective commands (except the unused ordinals, cf. [2], chapter 17)

	TPM_PROTECTED_ordinal +	Complete ordinal	AUTH2	AUTH1	RQU	Optional	No Owner	Physical Presence	PCR Use enforced	Audit	Duration	1.2 Changes	FIPS Changes	Avail Deactivated	Avail Disabled
TPM_ORD_ActivateIdentity	122	0x0000007A	X	X					X	X	M				
TPM_ORD_AuthorizeMigrationKey	43	0x0000002B		X						X	S				
TPM_ORD_CertifyKey	50	0x00000032	X	X	X				X		M				
TPM_ORD_CertifyKey2	51	0x00000033	X	X	X				X		M	N			
TPM_ORD_CertifySelfTest	82	0x00000052		X	X				X		M	X			
TPM_ORD_ChangeAuth	12	0x0000000C	X						X		M				
TPM_ORD_ChangeAuthAsymFinish	15	0x0000000F		X	X				X		M	D			
TPM_ORD_ChangeAuthAsymStart	14	0x0000000E		X	X				X		L	D			
TPM_ORD_ChangeAuthOwner	16	0x00000010		X					X	X	S				
TPM_ORD_CMK_ApproveMA	29	0x0000001D		X							S	N			
TPM_ORD_CMK_ConvertMigration	36	0x00000024		X					X		M	N			

	TPM_PROTECTED_ordinal +	Complete ordinal	AUTH2	AUTH1	RQU	Optional	No Owner	Physical Presence	PCR Use enforced	Audit	Duration	1.2 Changes	FIPS Changes	Avail Deactivated	Avail Disabled
TPM_ORD_CMK_CreateBlob	27	0x0000001B		X					X		M	N			
TPM_ORD_CMK_CreateKey	19	0x00000013		X					X		L	N	X		
TPM_ORD_CMK_CreateTicket	18	0x00000012		X							M	N			
TPM_ORD_CMK_SetRestrictions	28	0x0000001C		X			X	T			S	N			
TPM_ORD_ContinueSelfTest	83	0x00000053			X		X				L		X	X	X
TPM_ORD_ConvertMigrationBlob	42	0x0000002A		X	X				X	X	M				
TPM_ORD_CreateCounter	220	0x000000DC		X							S	N			
TPM_ORD_CreateEndorsementKeyPair	120	0x00000078			X		X				L				
TPM_ORD_CreateMaintenanceArchive	44	0x0000002C		X		X				X	S				
TPM_ORD_CreateMigrationBlob	40	0x00000028	X	X					X	X	M				
TPM_ORD_CreateRevocableEK	127	0x0000007F			X	X	X				L	N			
TPM_ORD_CreateWrapKey	31	0x0000001F		X					X	X	L		X		
TPM_ORD_DAA_Join	41	0x00000029		X		X					L	N			
TPM_ORD_DAA_Sign	49	0x00000031		X		X					L	N			
TPM_ORD_Delegate_CreateKeyDelegation	212	0x000000D4		X							M	N			
TPM_ORD_Delegate_CreateOwnerDelegation	213	0x000000D5		X							M	N			
TPM_ORD_Delegate_LoadOwnerDelegation	216	0x000000D8		X	X		X				M	N			
TPM_ORD_Delegate_Manage	210	0x000000D2		X	X		X				M	N			
TPM_ORD_Delegate_ReadTable	219	0x000000DB			X		X				S	N			
TPM_ORD_Delegate_UpdateVerification	209	0x000000D1		X							S	N			
TPM_ORD_Delegate_VerifyDelegation	214	0x000000D6			X						M	N			
TPM_ORD_DirRead	26	0x0000001A			X						S	D			
TPM_ORD_DirWriteAuth	25	0x00000019		X							S	D			
TPM_ORD_DisableForceClear	94	0x0000005E		X			X			X	S				
TPM_ORD_DisableOwnerClear	92	0x0000005C		X						X	S				
TPM_ORD_DisablePubekRead	126	0x0000007E		X						X	S				
TPM_ORD_DSAP	17	0x00000011			X						S	N		X	X
TPM_ORD_EstablishTransport	230	0x000000E6		X	X				X		S	N			
TPM_ORD_EvictKey	34	0x00000022			X						S	D			
TPM_ORD_ExecuteTransport	231	0x000000E7		X							? L	N			
TPM_ORD_Extend	20	0x00000014			X		X				S			X	X

	TPM_PROTECTED_ordinal +	Complete ordinal	AUTH2	AUTH1	RQU	Optional		No Owner	Physical Presence	PCR Use enforced	Audit	Duration	1.2 Changes	FIPS Changes	Avail Deactivated	Avail Disabled
TPM_ORD_FieldUpgrade	170	0x000000AA	X	X	X	X		X				?				
TPM_ORD_FlushSpecific	186	0x000000BA			X			X				S	N		X	X
TPM_ORD_ForceClear	93	0x0000005D			X			X	P		X	S				
TPM_ORD_GetAuditDigest	133	0x00000085			X	X		X			N	S	N			
TPM_ORD_GetAuditDigestsigned	134	0x00000086		X	X	X					N	M	N			
TPM_ORD_GetAuditEvent	130	0x00000082			X	X					N	S	X			
TPM_ORD_GetAuditEventsigned	131	0x00000083		X	X	X					N	M	X			
TPM_ORD_GetCapability	101	0x00000065			X			X				S	C		X	X
TPM_ORD_GetCapabilityOwner	102	0x00000066		X								S	D			
TPM_ORD_GetCapabilitySigned	100	0x00000064		X	X					X		M	X			
TPM_ORD_GetOrdinalAuditstatus	140	0x0000008C			X						N	S	X			
TPM_ORD_GetPubKey	33	0x00000021		X	X					X		S				
TPM_ORD_GetRandom	70	0x00000046			X			X				S				
TPM_ORD_GetTestResult	84	0x00000054			X			X				S			X	X
TPM_ORD_GetTicks	241	0x000000F1			X			X				S	N			
TPM_ORD_IncrementCounter	221	0x000000DD		X								S	N			
TPM_ORD_Init	151	0x00000097			X							M			X	X
TPM_ORD_KeyControlOwner	35	0x00000023		X								S	N			
TPM_ORD_KillMaintenanceFeature	46	0x0000002E		X		X					X	S				
TPM_ORD_LoadAuthContext	183	0x000000B7			X	X		X				M	D			
TPM_ORD_LoadContext	185	0x000000B9			X							M	N			
TPM_ORD_LoadKey	32	0x00000020		X	X					X		M	D	X		
TPM_ORD_LoadKey2	65	0x00000041		X	X					X		M	C	X		
TPM_ORD_LoadKeyContext	181	0x000000B5			X	X		X				S	D			
TPM_ORD_LoadMaintenanceArchive	45	0x0000002D		X		X					X	S				
TPM_ORD_LoadManuMaintPub	47	0x0000002F			X	X					X	S				
TPM_ORD_MakeIdentity	121	0x00000079	X	X						X	X	L		X		
TPM_ORD_MigrateKey	37	0x00000025		X	X					X		M	C			
TPM_ORD_NV_DefineSpace	204	0x000000CC		X	X			X	T			S	N		A	A
TPM_ORD_NV_ReadValue	207	0x000000CF		X	X			X	T*	X		S	N		A	A
TPM_ORD_NV_ReadValueAuth	208	0x000000D0		X					A*	X		S	N			
TPM_ORD_NV_WriteValue	205	0x000000CD		X	X			X	T*	X		S	N		A	A
TPM_ORD_NV_WriteValueAuth	206	0x000000CE		X					A*	X		S	N			

	TPM_PROTECTED_ordinal +	Complete ordinal	AUTH2	AUTH1	RQU	Optional	No Owner	Physical Presence	PCR Use enforced	Audit	Duration	1.2 Changes	FIPS Changes	Avail Deactivated	Avail Disabled
TPM_ORD_OIAP	10	0x0000000A			X		X				S			X	X
TPM_ORD_OSAP	11	0x0000000B			X						S			X	X
TPM_ORD_OwnerClear	91	0x0000005B		X						X	S				
TPM_ORD_OwnerReadInternalPub	129	0x00000081		X							S	C			
TPM_ORD_OwnerReadPubek	125	0x0000007D		X						X	S	D			
TPM_ORD_OwnerSetDisable	110	0x0000006E		X						X	S			X	X
TPM_ORD_PCR_Reset	200	0x000000C8			X		X				S	N		X	X
TPM_ORD_PcrRead	21	0x00000015			X		X				S				
TPM_ORD_PhysicalDisable	112	0x00000070			X		X	P		X	S			X	
TPM_ORD_PhysicalEnable	111	0x0000006F			X		X	P		X	S			X	X
TPM_ORD_PhysicalSetDeactivated	114	0x00000072			X		X	P		X	S			X	
TPM_ORD_Quote	22	0x00000016		X	X				X		M				
TPM_ORD_Quote2	62	0x0000003E		X	X	X			X		M	N			
TPM_ORD_ReadCounter	222	0x000000DE			X		X				S	N			
TPM_ORD_ReadManuMaintPub	48	0x00000030			X	X				X	S				
TPM_ORD_ReadPubek	124	0x0000007C			X		X			X	S				
TPM_ORD_ReleaseCounter	223	0x000000DF		X			X				S	N			
TPM_ORD_ReleaseCounterOwner	224	0x000000E0		X							S	N			
TPM_ORD_ReleaseTransportSigned	232	0x000000E8	X	X					X		M	N			
TPM_ORD_Reset	90	0x0000005A			X		X				S	C		X	X
TPM_ORD_ResetLockValue	64	0x00000040		X							S	N			
TPM_ORD_RevokeTrust	128	0x00000080			X	X	X	P			S	N			
TPM_ORD_SaveAuthContext	182	0x000000B6			X	X	X				M	D			
TPM_ORD_SaveContext	184	0x000000B8			X						M	N			
TPM_ORD_SaveKeyContext	180	0x000000B4			X	X	X				M	D			
TPM_ORD_SaveState	152	0x00000098			X		X				M			X	X
TPM_ORD_Seal	23	0x00000017		X					X		M				
TPM_ORD_Sealx	61	0x0000003D		X		X			X		M	N			
TPM_ORD_SelfTestFull	80	0x00000050			X		X				L			X	X
TPM_ORD_SetCapability	63	0x0000003F		X	X						S	N		X	X
TPM_ORD_SetOperatorAuth	116	0x00000074			X		X	P			S	N			
TPM_ORD_SetOrdinalAuditstatus	141	0x0000008D		X		X				X	S				
TPM_ORD_SetOwnerInstall	113	0x00000071			X		X	P		X	S				

	TPM_PROTECTED_ORDINAL +	Complete ordinal	AUTH2	AUTH1	RQU	Optional	No Owner	Physical Presence	PCR Use enforced	Audit	Duration	1.2 Changes	FIPS Changes	Avail Deactivated	Avail Disabled
TPM_ORD_SetOwnerPointer	117	0x00000075			X						S	N			
TPM_ORD_SetRedirection	154	0x0000009A			X	X		P		X	S				
TPM_ORD_SetTempDeactivated	115	0x00000073		X	X		X	O		X	S				
TPM_ORD_SHA1Complete	162	0x000000A2			X		X				S			X	X
TPM_ORD_SHA1CompleteExtend	163	0x000000A3			X		X				S			X	X
TPM_ORD_SHA1Start	160	0x000000A0			X		X				S			X	X
TPM_ORD_SHA1Update	161	0x000000A1			X		X				S			X	X
TPM_ORD_Sign	60	0x0000003C		X	X				X		M				
TPM_ORD_Startup	153	0x00000099			X		X				S			X	X
TPM_ORD_StirRandom	71	0x00000047			X		X				S				
TPM_ORD_TakeOwnership	13	0x0000000D		X			X			X	L			X	
TPM_ORD_Terminate_Handle	150	0x00000096			X		X				S	D		X	X
TPM_ORD_TickStampBlob	242	0x000000F2		X	X				X		M	N			
TPM_ORD_UnBind	30	0x0000001E		X	X				X		M				
TPM_ORD_Unseal	24	0x00000018	X	X					X		M	C			

The connection commands manage the TPM's connection to the TBB.

	TPM_PROTECTED_ORDINAL +	Complete ordinal	AUTH2	AUTH1	RQU	Optional	No Owner	PCR Use enforced	Audit	Duration	1.2	FIPS Changes	Avail Deactivated	Avail Disabled
TSC_ORD_PhysicalPresence	10	0x4000000A			X	X	X			S	C		X	X
TSC_ORD_ResetEstablishmentBit	11	0x4000000B			X	X	X			S	N		X	X

7.2. Acronyms

Acronym	Description
cmd	TPM command or commands as defined in [7]

Acronym	Description
DSAP	Delegate-Specific Authorization Protocol
EK	endorsement Key
HMAC	Keyed-Hashing for Message Authentication (cf. RFC 2104)
Mfg	Manufacturing (e.g. TPM-Mfg EK is the Endorsement key of the TPM generated during manufacturing)
NV	non-volatile (memory or area)
OIAP	Object-Independent Authorization Protocol
OSAP	Object-Specific Authorization Protocol
PCR	Platform Configuration Register
RTM	root of trust for measurement
RTR	root of trust for reporting
SAR	Security assurance requirement
SFR	security functional requirement
SRK	storage root key
TCG	Trusted Computing Group
TPM	Trusted Platform Module

7.3. Glossary

Term	Description
3DES	DES using a key of a size that is 3X the size that of a DES key. See DES.
Attestation	The process of vouching for the accuracy of information. External entities can attest to shielded locations, protected capabilities, and Roots of Trust. A platform can attest to its description of platform characteristics that affect the integrity (trustworthiness) of a platform. Both forms of attestation require reliable evidence of the attesting entity. [9]
Attestation Identity	An Attestation Identity Key (AIK) is an alias for the Endorsement Key.

Term	Description
Key (AIK)	The AIK is an asymmetric key pair used for signing PCR data only. For interoperability, the AIK is an RSA 2048-bit key.
AIK Credential	A credential issued by a Privacy Certification Authority (CA) that contains the public portion of an AIK key signed by a Privacy CA. The meaning and significance of the fields and the Privacy CA signature is a matter of policy. Typically it states that the public key is associated with a valid TPM. [9]
Authorization	In the TPM terminology: process of the identification, authentication and authorization of users by means of presented shared secrets (cf. [5], chapter 8).
Blob	Opaque data of fixed or variable size. The meaning and interpretation of the data is outside the scope and context of the Subsystem.
Context	A resource saved outside the TPM or loaded into the TPM (cf. [5], ch. 21, [6], ch. 18, [7] ch. 21)
Conformance Credential	A credential that vouches for the conformance of the TPM and the TBB to the TCG specifications.
Credential	Signed data containing information about public keys issued in the IT environment. Credential formats are expressed in ASN.1 notation and are expected to be able to leverage some elements of public key infrastructure. (cf. [11], sec. 4.2.5 for details).
DES	Symmetric key encryption using a key size of 56 bits defined by NIST as FIPS 46-3.
Direct Anonymous Attestation	A Protocol for vouching for an Attestation Identity Key (AIK) using zero-knowledge-proof technology. [9]
Endorsement Credential	A credential containing a public key (the endorsement public key) that was generated by a genuine TPM.
Endorsement Key (EK)	A term used ambiguously, depending on context, to mean a pair of keys, or the public key of that pair, or the private key of that pair; an asymmetric key pair generated by or inserted in a TPM that is used as proof that a TPM is a genuine TPM; the public endorsement key (PUBEK); the private endorsement key (PRIVEK).
Identity Credential	A credential for an Attestation Identity Key issued by a Privacy CA that provides an identity for the TPM.
Integrity metric(s)	Values that are the results of measurements on the integrity of the platform.
HMAC	keyed-hashing message authentication code according to RFC 2104
Man-in-the-middle attack	An attack by an entity intercepting communications between two others without their knowledge and by intercepting that communication is able to obtain or modify the information between

Term	Description
	communication is able to obtain or modify the information between them.
Migratable	A key which may be transported outside the specific TPM.
Nonce	A nonce is a random value that provides protection from replay and other attacks. Many of the commands and protocols in the specification require a nonce.
Non-Migratable	A key which cannot be transported outside a specific TPM; a key that is (statistically) unique to a particular TPM.
operator	Anyone who has physical access to a platform [9].
Owner	The entity that owns the platform in which a TPM is installed. Since there is, by definition, a one-to-one relationship between the TPM and the platform, the Owner is also the Owner of the TPM. The Owner of the platform is not necessarily the “user” of the platform (e.g., in a corporation, the Owner of the platform might be the IT department while the user is an employee.) The Owner has administration rights over the TPM.
Payload of TPM command	In the context of transport protection: the data of a TPM command except the ordinal, the header information, keys, handles and autorizations which are encrypted in a warped transport command, cf. [5], sec. 8.1, for details.
PKI Identity Protocol	The protocol used to insert anonymous identities into the TPM.
Platform Credential	A credential that states that a specific platform contains a genuine TCG Subsystem.
Privacy CA	An entity that issues an Identity Credential for a TPM based on trust in the entities that vouch for the TPM via the Endorsement Credential, the Conformance Credential, and the Platform Credential.
Private Endorsement Key (PRIVEK)	The private key of the key pair that proves that a TPM is a genuine TPM. The PRIVEK is (statistically) unique to only one TPM.
Public Endorsement Key (PUBEK)	A public key that proves that a TPM is a genuine TPM. The PUBEK is (statistically) unique to only one TPM.
Random number generator (RNG)	A pseudo-random number generator that must be initialised with unpredictable data and provides, “random” numbers on demand.
Root of Trust for Measurement (RTM)	The point from which all trust in the measurement process is predicated.
Root of Trust for Reporting (RTR)	The point from which all trust in reporting of measured information is predicated.

Term	Description
Root of Trust for Storing (RTS)	The point from which all trust in Protected Storage is predicated.
RSA	An (asymmetric) encryption method using two keys a private key and a public key. Reference http://www.rsa.com .
SHA-1	A NIST defined hashing algorithm producing a 160-bit result from an arbitrary sized source as specified in FIPS 180-2.
Storage Root Key (SRK)	The root key of a hierarchy of keys associated with a TPM; generated within a TPM; a non-migratable key.
TPM Identity	One of the anonymous PKI identities belonging to a TPM; a TPM may have multiple identities bind to an Attestation Identity Key.
TPM-protected capability	A function which is protected within the TPM, and has access to TPM secrets.
Transport log of commands	Hash value of command parameters of a transport session generated by the commands TPM_EstablishTransport, TPM_ExecuteTransport and TPM_ReleaseTransportSigned and is signed and returned by the command TPM_ReleaseTransportSigned.
Trusted Building Block (TBB)	The parts of the Root of Trust that do not have shielded locations or protected capabilities. Normally includes just the instructions for the RTM and the TPM initialization functions (reset, etc.). Typically platform-specific. One example of a TBB is the combination of the CRTM, connection of the CRTM storage to a motherboard, the connection of the TPM to a motherboard, and mechanisms for determining Physical Presence. [9]
Trusted Platform Module (TPM)	The set of functions and data that are common to all types of platform, which must be trustworthy if the Subsystem is to be trustworthy; a logical definition in terms of protected capabilities and shielded locations.
Trusted Platform Support Services	The set of functions and data that are common to all types of platform, which are not required to be trustworthy (and therefore do not need to be part of the TPM).
User	An entity that uses the platform in which a TPM is installed. The only rights that a User has over a TPM are the rights given to the User by the Owner. These rights are expressed in the form of authentication data, given by the Owner to the User, that permits access to entities protected by the TPM. The User of the platform is not necessarily the "owner" of the platform (e.g., in a corporation, the owner of the platform might be the IT department while the User is an employee). There can be multiple Users.
Validation Credential	A credential that states values of measurements that should be obtained when measuring a particular part of the platform when the part is functioning as expected.

Term	Description
Validation Data	Data inside a Validation Credential; the values that the integrity measurements should produce when the part of a platform described by the Validation Credential is working correctly.
Validation Entity	An entity that issues a Validation Certificate for a component; the manufacturer of that component; an agent of the manufacturer of that component.

7.4. Literature

- [1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 1, CCMB-2006-09-001, September 2006
- [2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 3.1, Revision 2, CCMB-2007-09-002, September 2007
- [3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 2, CCMB-2007-09-003, September 2007
- [4] Common Methodology for Information Technology Security Evaluation Methodology, Evaluation Methodology, Version 3.1, Revision 2, CCMB-2007-09-004, September 2007
- [5] TPM Main Part 1 Design Principles, Specification Version 1.2, Revision 94, 29 March 2005, Trusted Computing Group, Incorporated
- [6] TPM Main Part 2 TPM Structures, Specification Version 1.2, Revision 94, 29 March 2005, Trusted Computing Group, Incorporated
- [7] TPM Main Part 3 Commands, Specification Version 1.2, Revision 94, 29 March 2005, Trusted Computing Group, Incorporated
- [8] TCG PC Client Specific TPM Interface Specification (TIS) for TPM Family 1.2; Level 2, Version 1.2 FINAL, Revision 1.00, July 11, 2005
- [9] TCG Glossary of Technical Terms,
<https://www.trustedcomputinggroup.org/groups/glossary/>
- [10] Trusted Computing Platform Alliance (TCPA) Trusted Platform Module Protection Profile, Version 1.9.7, July 1, 2002
- [11] TCG Specification Architecture Overview, Specification, Revision 1.4, 2nd August 2007
- [12] IEEE P1363-2000, Standard Specifications for Public Key Cryptography, Institute of Electrical and Electronics Engineers, Inc. (note reaffirmation PAR is actual running)

- [13] FIPS PUB 180-2 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, SECURE HASH STANDARD, National Institute of Standards and Technology, 2002 August 1
- [14] RFC 2104: HMAC: Keyed-Hashing for Message Authentication, <http://www.ietf.org/rfc/rfc2104.txt>
- [15] PKCS #1 v2.0: RSA Cryptography Standard, RSA Laboratories, October 1, 1998
- [16] Federal Information Processing Standards Publication 197: Specification for the ADVANCED ENCRYPTION STANDARD (AES), November 26, 2001
- [17] FIPS PUB 46-3 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION DATA ENCRYPTION STANDARD (DES) Reaffirmed 1999 October 25
- [18] NIST Special Publication 800-17: Modes of Operation Validation System (MOVS): Requirements and Procedures, February 1998

8. Optional Package “Revoke of Trust” (Informative Annex)

The TPM specification describes optional functions which may be mandatory for TPM on specific platforms or may be supported by TPM manufacturer. This informative annex to the protection profile provides a functional package to support the ST writer to describe security functional requirements for revoke of trust.

Platform trust is demonstrated using the Endorsement Key Credential, the Platform Credential and the Conformance Credentials. There are circumstances where clearing all keys and values within the TPM is either desirable or necessary. The deleting of the Endorsement Key deletes all non-migratable keys and owner-specified state, and invalidates the Endorsement Key Credential and the Platform Credential. The TPM shall control the loading, creating and deleting the Endorsement Key. The Endorsement Key Credential shall indicate whether the Endorsement Key was squirted or internally generated, and revocable or non-revocable. The Endorsement Key may be configured to be revocable and may be substituted by a new one.

The ST writer may introduce a new object Revocable EK (instead of the object #1 EK in table 1), the new organizational security policy OSP.Trust and the security objective O.Trust_Management.

#	Object	Operation	Security attributes and authorization data
14	<p>Revocable EK</p> <p>The revocable TPM Endorsement Key (EK) is an asymmetric RSA key pair which identifies uniquely each TPM device but may be revoked and a new EK may be generated.</p>	<ul style="list-style-type: none"> - create: create a revocable EK (cf. cmd TPM_CreateRevocableEK [7], sec. 14.2)²¹⁶. - use: take ownership (cf. cmd TPM_TakeOwnership, [7], sec. 6.1), decryption of AIK credentials (cf. cmd TPM_ActivateIdentity, [7], sec. 15.2), prove a DAA attestation held by a TPM (cf. cmd TPM_DAA_Sign, [7], sec. 26.2) - read: exports the public portion of the EK (cf. cmd TPM_ReadPubek [7], sec. 14.4, TPM_OwnerReadInternalPub 	<p>ownerAuth: authorization data to read public key part and to use EK, defined in TPM_PERMANENT_DATA (cf. [6], sec. 7.4)</p> <p>enableRevokeEK: if TRUE than the TPM_RevokeTrust command is active; if FALSE than the TPM RevokeTrust command is disabled (cf. TPM_PERMANENT_FLAGS, [6], sec. 7.1)</p> <p>ekReset: authorization data for TPM_RevokeTrust (cf. TPM_PERMANENT_DATA,</p>

²¹⁶ Note the TPM specification [5] and [7] allows for the manufacturing option to generate the EK outside in the IT environment and to “squirt” the EK into the TOE. If this option is used for the TOE the ST shall state these processes and the evaluation will assess the security of these process in the evaluation sub-activities ALC_DVS.1 Identification of security measures and ALC_DEL.1 Delivery procedures.

#	Object	Operation	Security attributes and authorization data
		<p>[7], sec. 14.5)</p> <ul style="list-style-type: none"> - delete: delete the EK (cf. cmd TPM_RevokeTrust [7], sec. 14.3) 	<p>[6], sec. 7.4,</p> <p>CEKPUSED: if TRUE: than the PRIVEK and PUBEK were created using TPM_CreateEndorsement-KeyPair; if FALSE than the PRIVEK and PUBEK were created using a manufacturers process (EK is "squirted" or generated by the TPM (cf. TPM_PERMANENT_FLAGS, [6], sec. 7.1)</p> <p>readpubEK: export of public portion allowed (cf. in TPM_PERMANENT_FLAGS, [6], sec. 7.1)</p>

#	OSP	Description
8	OSP.Trust	Platform trust is demonstrated using the Endorsement Key Credential, the Platform Credential and the Conformance Credentials. Deleting of the Endorsement Key deletes all non-migratable keys and owner-specified state, and invalidates the Endorsement Key Credential and the Platform Credential. The Endorsement Key Credential shall indicate whether the Endorsement Key was squirted or internally generated, and revocable or non-revocable.

#	Objective	Description
21	O.Trust_Management	The TPM shall control the creation and deletion of the revocable Endorsement Key.

The ST writer may describe the security functional requirements for revocation of trust as follows.

FCS_CKM.1/REK Cryptographic key generation – Revocable Endorsement Key

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/REK The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *RSA*²¹⁷ and specified cryptographic key sizes *2048 bit*²¹⁸ that meet the following: *P1363*²¹⁹.

FDP_ACC.1/REK Subset access control– Revocable Endorsement Key

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

FDP_ACC.1.1/REK The TSF shall enforce the *Revoke EK SFP*²²⁰ on

- (1) *Subjects: TPM owner,*
- (2) *Objects: Revocable EK,*
- (3) *Operations: command TPM_CreateRevocableEK*²²¹.

FDP_ACF.1/REK Security attribute based access control– Revocable Endorsement Key

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

FDP_ACF.1.1/REK The TSF shall enforce the *Revoke EK SFP*²²² to objects based on the following:

- (1) *Subjects: TPM owner with security attribute physical presence,*
- (2) *Objects: Revocable EK with security attributes enableRevokeEK, EKReset, CEKPUse*²²³.

²¹⁷ [assignment: *cryptographic key generation algorithm*]

²¹⁸ [assignment: *cryptographic key sizes*]

²¹⁹ [assignment: *list of standards*]

²²⁰ [assignment: *access control SFP*]

²²¹ [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

²²² [assignment: *access control SFP*]

²²³ [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/REK The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- (1) *The TSF shall allow to delete the Revocable EK if*
 - (a) *the command TPM_CreateRevocableEK is executed under physical presence,*
 - (b) *the parameter EKReset in the command matches value EKReset stored in the TPM_PERMANENT_DATA,*
 - (c) *the security attribute enableRevokeEK is TRUE*
- (2) *The TPM owner under physical presence is allowed to create the Revocable EK if*
 - (a) *the security attributes enableRevokeEK equal TRUE and*
 - (b) *CEKPUse equal TRUE and the EK does not exist²²⁴.*

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

The security objective rationale will map the OSP.Trust to O.Trust_Management. The security requirements rationale will show that O.Trust_Management is covered by FCS_CKM.1/REK, FDP_ACC.1/REK, and FDP_ACF.1/REK.

²²⁴ [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]