Protection Profile
Automotive-Thin Specific TPM

TCG TPM 2.0 Automotive Thin Profile
Family "2.0"
Level 0 Version 1.0
12 December 2018

Contact: admin@trustedcomputinggroup.org

TCG Published

TCG

Disclaimers, Notices, and License Terms

Table of Contents

Tables

This page is intentionally left blank.

Revision History

Revision	Date	Description
1.0	12.12.2018	First official release

1. Scope

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

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

1.1 Key words

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

1.2 Statement Type

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

2. Protection Profile (PP) Introduction

2.1 PP Reference

Title: Protection Profile Automotive-Thin Specific Trusted Platform Module Specification Family 2.0; Level 0; Version 1.0 (PP AutoThin TPM F2.0 L0 V10)

Sponsor: Trusted Computing Group

CC Version: 3.1 (Release 5)

Assurance level: EAL4 augmented with ALC_FLR.1 and AVA_VAN.4

Document version: 1.0

Keywords: trusted computing group, trusted platform module, Automotive-Thin specific TPM

2.2 TOE Overview

2.2.1 TOE Definition

The TOE is the TCG Automotive-Thin Specific Trusted Platform Module (ATS TPM). This ATS TPM is a device that implements the functions defined in the TCG Trusted Platform Module Library Specification, version 2.0, [7], [8], [9], [10], and the TCG TPM 2.0 Automotive Thin Profile [32]. The TCG Trusted Platform Module Library specification describes the design principles, the TPM structures, the TPM commands and supporting routines for the commands. The TPM Automotive-Thin specific interface specification describes the additional features that must be implemented by an ATS TPM for automotive electronic control unit (ECU) platform.

The TOE consists of

(1) TPM hardware,

(2) TPM firmware,

(3) TPM guidance documentation.

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

2.2.2 TOE Usage and Security Features

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

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

The TPM provides methods for collecting and reporting identities of hardware and software components of an automotive platform. The automotive system report generated by the Trusted Computing Base (TCB) that the TPM is part of allows determination of expected behavior, and from that, allows an expectation of trust in the automotive system platform.

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

Support for the Root of Trust for Measurement

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

Root of Trust for Reporting

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

Root of Trust for Storage

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

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

Platform Key Hierarchy

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

Other Security Services and Features

The TOE provides cryptographic services for hashing, asymmetric encryption and decryption, asymmetric signing and signature verification, symmetric encryption and decryption, symmetric signing and signature verification and key generation. Hash function SHA-256 is provided as a cryptographic service to external entities for measurements and used internally for user authentication, signing and key derivation. A TOE is required to implement asymmetric algorithms: where the current specification supports RSA with 2048 bits for digital signature, secret sharing and encryption or ECC algorithms with P-256 curves for digital signatures and secret sharing. The TOE provides symmetric encryption and decryption of AES-128 in CFB mode. The TOE implements symmetric signing and signature verification by means of HMAC described in [15]. The TOE generates two types of keys: Ordinary keys are generated using the random number generator to seed the key computation. Primary Keys are derived from a Primary Seed and key parameters by means of a key derivation function.

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

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

2.2.3 Non-TOE Hardware, Firmware and Software

The TPM is a hardware component of an automotive ECU platform. The ECU Platform firmware interacts with the TPM by sending commands to the TPM and receiving responses from the TPM through the interface described in [7] and [32]. Further, the TPM is able to obtain the indication _TPM_Init and adjust its internal state accordingly. Therefore, the TOE is a passive device controlled by the software running on the ECU platform.

A modern automotive vehicle typically has over 100 separate processors (each with its own OS, RAM, and applications) that are called Electronic Control Units (ECUs) and are configured on three or more separate and isolated networks as shown in Figure 1. Even though the automotive vehicle appears to be a single object that is Internet-connected, the vehicle is actually a complex system of separate networks that includes a Head-Unit or Gateway communicating with a Remote Center (a vehicle safety and maintenance center, typically operated by a manufacturer or government agency). The Head Unit or Gateway communicates on behalf of ECUs that face constraints imposed by the demands of high performance real-time machines, and performance requirements from a driver, passengers, and outside highway environmental factors (e.g., road conditions, traffic density, lighting, weather, etc.).

Given the diverse use cases inside the vehicle, it is reasonable to describe a vehicle as a composite industrial control system network with one or more Internet Gateways and one or more human user interfaces. Due to the complexity of this automotive vehicle model, the Automotive Library Profile [32] is limited to a definition of the functionality of a TPM that can be deployed in each resource-constrained ECU within the vehicle.

Significant characteristics of an Automotive-Thin TPM include:

1. Often deployed in support of resource-constrained ECUs to support their integrity and attestation for remote maintenance services

2. Supports storage of ECU firmware measurements, creation of integrity digests, and creations of signatures on integrity digests

3. After receiving and installing a firmware update or patch, an ECU may use an Automotive-Thin TPM to help provide confirmation that an update installation was completed successfully.

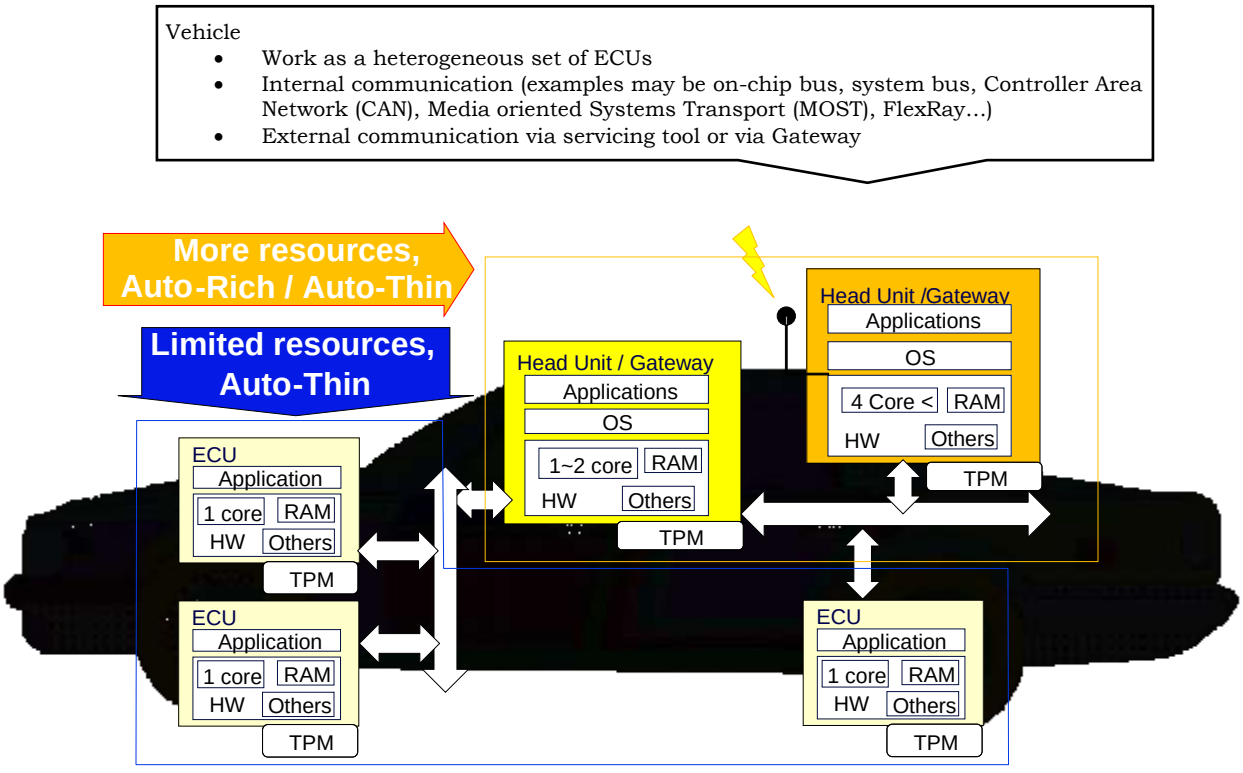


Figure 1: Overview of an Automotive vehicle using TPM technology

For the case where each ECU has its own Automotive-Thin TPM, the number of Automotive-Thin TPMs may be over 100. This is the reason the Automotive-Rich Profile could store copies of individual Automotive-Thin PCRs in its own NVRAM and also aggregate the integrity measurements from the many Automotive-Thin TPMs.

The current Protection Profile only addresses the Automotive-Thin TPM. This PP addresses all mandatory commands and some recommended commands for the Automotive-Thin TPM. The two field upgrade commands, TPM2_FieldUpgradeStart and TPM2_FieldUpgradeData are recommended and covered in the PP-Module in Appendix 8. Please refer to the Automotive-Thin Profile document [33] for more details on the message flows between Remote Center and ECU for remote maintenance of ECUs (ECU firmware updates) and for the list of mandatory, recommended and optional commands.

2.2.4 TPM Life Cycle

The TPM life cycle may be described in four phases: Development, Manufacturing, Platform Integration and Operational usage. The TPM life cycle distinguishes one case,

- Case 1: The TPM hardware and TPM firmware are manufactured and delivered together.

Because the current Automotive Thin TPM supports optional Field Upgrade, the TPM life cycle distinguishes a second case, defined in an optional package in Appendix 8,

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

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

- Development of the TPM (Phase 1)

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

- Manufacturing and Delivery of the TPM (Phase 2)

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

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

  This phase ended with TPM delivery to the customer.

- Platform Integration (Phase 3)

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

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

- Operational usage (Phase 4)

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

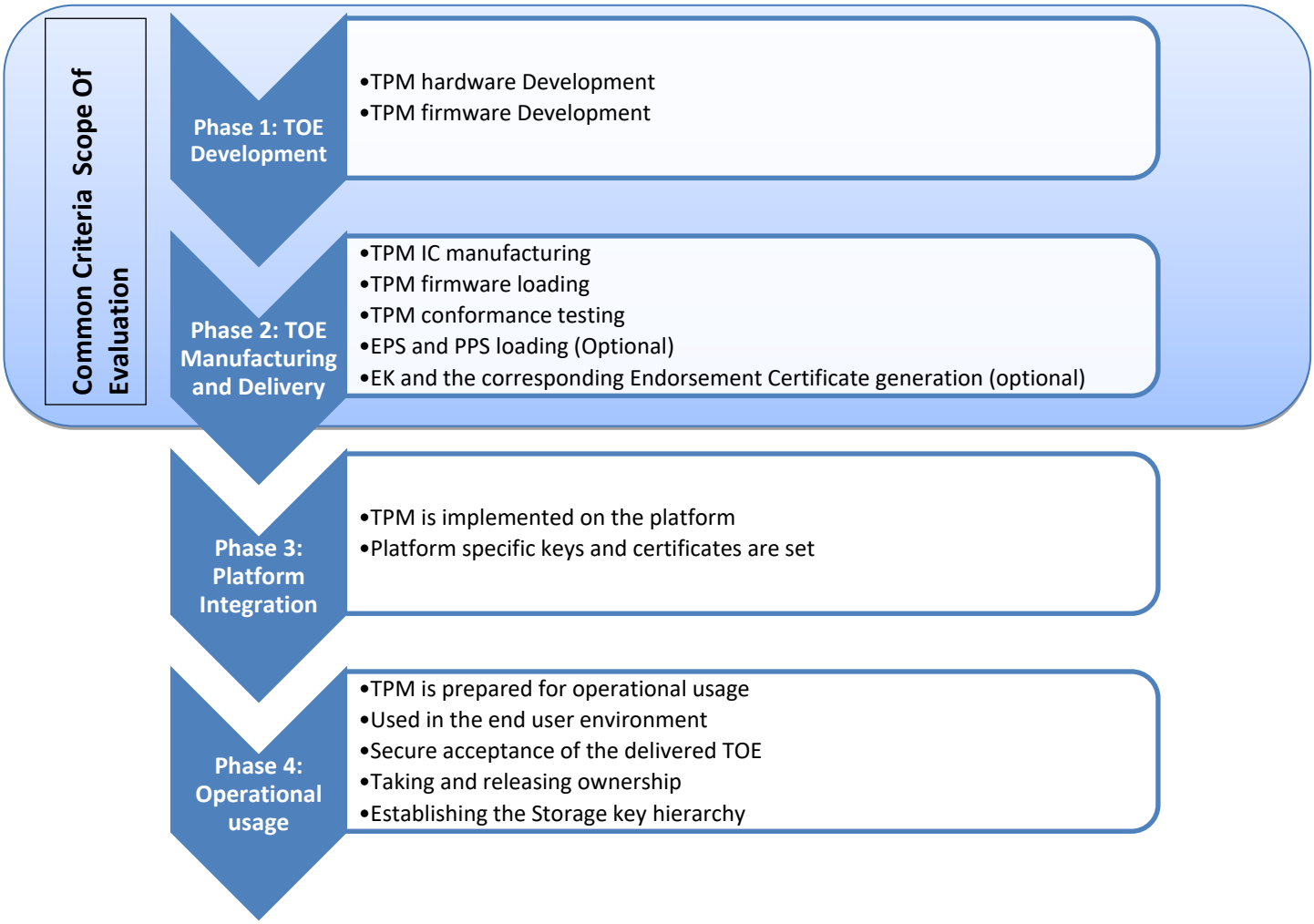


Figure 2: TPM Life Cycle case 1

The Common Criteria evaluation covers the Development of the TOE (Phase 1), the Manufacturing of the TPM (phase 2) up to the delivery to the platform vendor under the development environment (cf. CC part 1 [1], paragraph 157) in the evaluator activity of class ALC: Life-cycle support. The concrete state of the TPM when delivered to the platform vendor as customer of the TPM vendor depends on the vendor configuration options. A TPM can be delivered with no key, or with an Endorsement Key, or with an Endorsement Key

and an Endorsement Certificate, or with a Platform Key and a Platform Certificate. The security target shall describe all configurations of the TOE as delivered to the platform vendor. Details on these configurations will be provided for evaluator activities of families ALC_CMS and ALC_DEL. The user guidance provided by the TPM vendor shall describe the requirement and general procedures and the supplier of the certified TOE shall obey these procedures enabling the end users acceptance of certified version and configuration of the delivered TOE. (cf. element AGD_PRE.1.1C for details).

3. Conformance Claims

The following sections describe the conformance claims of the Protection Profile Automotive-Thin Specific Trusted Platform Module.

3.1 CC Conformance Claim

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

- Part 2 extended,
- Part 3 conformant.

3.2 Conformance with Packages

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

3.3 Conformance with other Protection Profiles

This PP does not claim conformance to any other PP.

3.4 Conformance Statement

This PP requires **strict** conformance of any Security Target (ST) or PP that claims conformance to this PP.

4. Security Problem Definition

The following sections describe the security problem definition of the Protection Profile for the Automotive-Thin Specific Trusted Platform Module.

4.1 Assets

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

The assets are:

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

4.2 Threats

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

Table 1: Threats

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

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

4.3 Organisational Security Policies (OSP)

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

Table 2: Organisational Security Policies

#	OSP	Description
1	OSP.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.
2	OSP.RT_Reporting	The root of trust for reporting reports on the contents of the RTS. A RTR report is typically a digitally signed digest of the contents of selected values within a TPM (measurement, key properties or audit digest). The authenticity of the assets reported is based on the verification of the signature and the certificate of the signing key.
3	OSP.RT_Storage	The root of trust for storage protects the assets (listed in Table 8 and Table 9) entrusted to the TPM in confidentiality and integrity.

4.4 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 3: Assumptions about the IT Environment

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

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 piecewise solution is called the security objectives for the TOE and consists of a set of statements describing the security goals that the TOE should achieve in order to solve its part of the problem.

Table 4: Security Objectives for the TOE

#	Objective	Description
1	O.Crypto_Key_Man	The TOE must manage cryptographic keys, including generation of cryptographic keys using the TOE random number generator as a source of randomness, in a manner to protect their confidentiality and integrity.
2	O.DAC	The TOE must control and restrict user access to the TOE protected capabilities and shielded locations in accordance with a specified access control policy where the object owner manages the access rights for their data objects using the principle of least privilege.
3	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.
4	O.Fail_Secure	The TOE must enter a secure failure mode in the event of a failure.
5	O.General_Integ_Checks	The TOE must provide checks on system integrity and user data integrity.
6	O.I&A	The TOE must identify all users, and shall authenticate the claimed identity except that of the role “World” before granting a user access to the TOE facilities.
7	O.Import	When data are being imported into the TOE, the TOE must ensure that the data security attributes are being imported with the data and the data is from an authorised source. In addition, the TOE shall verify those security attributes according to the TSF access control rules. The TOE supports the protection of confidentiality and the verification of the integrity of imported data.
8	O.Limit_Actions_Auth	The TOE must restrict the actions a user may perform before the TOE verifies the identity of the user.

#	Objective	Description
9	O.Record_Measurement	The TOE must support calculating hash values and recording the result of a measurement.
10	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.
11	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.
12	O.Reporting	The TOE must report measurement digests and attest to the authenticity of measurement digests.
13	O.Security_Attr_Mgt	The TOE must allow only authorised users to initialise and to change security attributes of objects and subjects. The management of security attributes shall support the principle of least privilege by means of role based administration and separation of duty.
14	O.Security_Roles	The TOE must maintain security-relevant roles and association of users with those roles.
15	O.Self_Test	The TOE must provide the ability to test itself, verify the integrity of the shielded data objects and that the protected capabilities operate as designed and enter a secure state in the case of detected errors.
16	O.Single_Auth	The TOE must provide a mechanism to authenticate a single user and require re-authentication to prevent “replay” and “man-in-the-middle” attacks.
17	O.Sessions	The TOE must provide the confidentiality of the parameters of the commands within an authorised session and the integrity of the audit log of the commands.
18	O.Tamper_Leak_Resistance	The TOE must resist physical tampering of the TSF by hostile users. The TOE must protect assets against leakage.

5.2 Security Objectives for the Operational Environment

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

Table 5: Security Objectives for the Operational Environment

#	Objective Name	Objective Description
1	OE.Configuration	The TOE must be installed and configured properly for starting up the TOE in a secure state. The security attributes of subjects and objects shall be managed securely by the authorised user.
2	OE.Credential	The IT environment must create EK credentials by trustworthy procedures for the root of trust for reporting.
3	OE.Measurement	The platform part of the root of trust for measurement provides a representation of embedded data or program code (measured values) to the TPM for measurement.

5.3 Security Objective Rationale

The following table provides an overview of the mapping between the security objective for the TOE and the functional security requirements. The table shows and the rationale demonstrates that each security objective for the TOE is traced back to threats countered by that security objective and OSPs enforced by that security objective; each security objective for the operational environment is traced back to threats countered by that security objective, to OSPs enforced by that security objective, and to assumptions upheld by that security objective. All security objectives counter all threats, enforce all organisational security policies and uphold all assumptions.

Table 6: Security Objective Rationale

	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.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Leak_Resistance	OE.Configuration	OE.Credential	OE.Measurement
T.Compromise		X				X					X			X							
T.Bypass													X	X							
T.Export			X										X						X		
T.Hack_Crypto	X																				
T.Hack_Physical		X																X			
T.Imperson						X	X	X						X							
T.Import							X														

	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.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Leak_Resistance	OE.Configuration	OE.Credential	OE.Measurement
T.Insecure_State				X	X								X						X		
T.Intercept			X				X										X				
T.Malfunction				X											X						
T.Modify		X				X		X						X							
T.Object_Attr_Change													X								
T.Replay																X					
T.Repudiate_Transact										X											
T.Residual_Info											X										
T.Leak																		X			
OSP.RT_Measurement									X												X
OSP.RT_Reporting												X								X	
OSP.RT_Storage	X	X	X			X	X														
A.Configuration																		X			

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

T.Imperson is countered by O.I&A, O.Security_Roles, O.Import, O.Limit_Actions_Auth. These objectives prevent impersonation by authentication based on managed roles with their associated 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 that of the role "World" before granting a user access to the TOE facilities. This objective provides the prerequisite for the application of the access control roles for the subjects by uniquely identifying users and requiring authentication of the user bound to a subject.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 as follows.

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

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

OSP.RT_Reporting: The root of trust for reporting reports on the contents of the RTS. A RTR report is typically a digitally signed digest of the contents of selected values within a TPM (measurement, key properties or audit digest). The authenticity of the assets reported is based on the verification of the signature and the credential of the signing key.

The OSP.RT_Reporting is implemented by the objectives

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

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

OSP.RT_Storage: The TPM as root of trust for storage protects the assets (listed in Table 8 and Table 9) entrusted to the TPM in confidentiality and integrity.

The 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 the hierarchy of Primary Objects that are stored outside of the TOE:

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

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

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

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

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

the principle of least privilege. This objective addresses the access control for the objects.

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

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

6. Extended Components Definition

This protection profile defines the extended Security Functional Requirement (SFR):

- Family Random Number Generation (FCS_RNG) of the class FCS: cryptographic support in order to describe the generation of random numbers for cryptographic purposes.

6.1 Family Random Number Generation

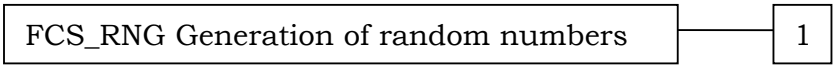
The family Random Number Generation (FCS_RNG) of the class FCS: cryptographic support describes the security functional requirements for random number generation used for cryptographic purposes. Random number generation is provided to the user and used internally, but it is not limited to generation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

FCS_RNG Generation of random numbers

Family behavior

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

Component leveling:



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

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

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

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

7. Security Requirements

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

7.1 Security Functional Requirements

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

7.1.1 Definitions of Subjects, Objects and TSF data

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

Table 7: Subjects

Subject	Description	TSF data
Platform firmware	Entity that controls the platform hierarchy	platformAuth, security attributes: physical presence if supported by the TOE[1]
Privacy administrator	Entity that controls the endorsement hierarchy	endorsementAuth,
USER	Entity that uses objects, keys, data in NV memory	authValue, assigned to the object
World	Entity not authenticated	(none)

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

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

#	Protected Objects	Operations	Security attributes
1	**Platform Hierarchy** Set of services to manage Platform firmware controls	**Seed** The PPS may be installed at manufacturing time or generated automatically on first boot **Change authorisation** (cmd	Authorisation data: **platformAuth**, hierarchy authorisation to change platform policy or authorisation and disable the platform hierarchy. Security attributes: **hierarchy proof**, secret value used

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

#	Protected Objects	Operations	Security attributes
		TPM2_HierarchyChangeAuth)	to associate a hierarchy with tickets, objects or contexts
2	**Endorsement Hierarchy** Set of services to manage Privacy Administrator controls	**Seed** The EPS may be installed at manufacturing time or generated automatically on first boot **Change authorisation** (cmd TPM2_HierarchyChangeAuth)	Authorisation data: **platformAuth**, hierarchy authorisation to enable/disable the Endorsement hierarchy. **endorsementAuth**, hierarchy authorisation to change the authorisation for the Endorsement hierarchy. Security attributes: **hierarchy proof**, secret value used to associate a hierarchy with tickets, objects or contexts
3	**Platform Primary Object** A root key created by the TPM that may be stored in the TPM or cached outside the TPM. The resource is instantiated in the Platform Hierarchy.	**Create** (cmd TPM2_CreatePrimary) **Delete** (cmd TPM2_EvictControl) **Make Persistent** (cmd TPM2_EvictControl)	Authorisation data: **authValue,** User auth secret value for the primary key **platformAuth**, hierarchy authorisation to use the Platform Primary Seed. Security attributes: **key template**, TPMT_Public, Part 2, §12.2.4, the public parameters used to create the key, set by the cmd TPM2_CreatePrimary
4	**Endorsement Primary Key** A root key created by the TPM that may be stored in the TPM or cached outside the TPM. The resource is instantiated in the Endorsement Hierarchy.	**Create** (cmd TPM2_CreatePrimary) **Delete** (cmd TPM2_EvictControl) **Make Persistent** (cmd TPM2_EvictControl)	Authorisation data: **authValue,** User auth secret value **platformAuth**, hierarchy authorisation to use the Platform Primary Seed. **endorsementAuth**, hierarchy authorisation to use the Endorsement Primary Seed. Security attributes: **key template**, TPMT_Public, Part 2, §12.2.4, the public parameters

#	Protected Objects	Operations	Security attributes
			used to create the key, set by the cmd TPM2_CreatePrimary
5	**User Key** Any cryptographic key except the primary keys.	**Create** (cmd TPM2_Create) **Make Persistent** (cmd TPM2_EvictControl) **Load** (cmd TPM2_Load) **Delete** (cmd TPM2_EvictControl)	Authorisation data: **authValue,** User auth secret value **platformAuth**, hierarchy authorisation to use the Platform Primary Object. **endorsementAuth**, hierarchy authorisation to use the Endorsement Primary Key. Security attributes: **key template**, TPMT_Public, Part 2, §12.2.4, the public parameters used to create the key, set by the cmd TPM2_Create
6	**PCR** Platform Configuration Register (PCR) intended to record measurement digests and to be used for attestation and access control.	**reset**: set all PCR to their default initial condition or to their save state (cmd TPM2_Startup) **read**: read the value of all PCRs specified in pcrSelect (cmd TPM2_PCR_Read), **quote**: hash the selected PCR, sign the value with an identified signing key and export it (cmd TPM2_Quote) **extend**: calculate the hash value of the PCR value according to the digests list or the result of a pending hash calculation (cmd TPM2_PCR_ Extend). **event:** calculate the hash value of the eventData and return the digests list, in case an implemented PCR is referenced, and an extend of the digests list is processed	Authorisation data: **authValue** Security attributes: All flags are defined in [8], sec. 6.14 TPM_PT_PCR TPM_PT_PCR_SAVE - indicates that the PCR is saved and restored by TPM_SU_STATE

#	Protected Objects	Operations	Security attributes
		(cmd TPM2_PCR_Event)	
7	**NV storage** Non-volatile storage of the TPM provided to the user and protected by access rights managed by the TPM owner.	TPM2_NV_Read TPM2_NV_ReadPublic TPM2_EvictControl	**TPM_NV_INDEX** Security attributes: platform controls (TPMA_NV_PPWRITE and TPMA_NV_PPREAD) user controls (TPMA_NV_AUTHREAD and TPMA_NV_AUTHWRITE) additional security attributes: cf. [8], sec. 13.2, table 196
8	**RNG** The TPM random number generator (RNG) creates random numbers provided to the user and for internal use (e.g. key generation, secrets, nonce).	**read**: read the next random number generated by the TPM (cf. cmd TPM2_GetRandom), **refresh**: provides any data as input to the random number generator to refresh the internal state of the random number generator (cf. cmd TPM2_StirRandom)	No security attributes

7.1.2 Presentation of operations on SFR components

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, the corresponding words are crossed out ~~like this.~~

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

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the values of security attributes. Assignments that have been made by the PP authors are denoted by showing as underlined text and the original text of the component is given by a footnote. Assignments to be filled in by the ST author appear in square

brackets with an indication that an assignment is to be made [assignment:], and are *italicised*. If assignment is performed but requires further selection or assignment, the operation is printed as underlined text like this [selection:] or [assignment:], and the open operation is printed *italicised and underlined*.

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

7.1.3 SFRs for the General Behavior of the TOE

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

7.1.3.1 Management

FMT_SMR.1 Security roles

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

FMT_SMR.1.1 The TSF shall maintain the roles

(1) Platform firmware,
(2) Privacy Administrator,
(3) USER,
(4) World[2].

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

Application note 1: The roles Platform firmware and Privacy Administrator are defined for the hierarchies. The role USER is defined for objects.

FMT_SMF.1 Specification of Management Functions

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

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

(1) Management of hierarchies,
(2) Management of authorisation values,
(3) Management of security attributes of keys,
(4) Management of security attributes of PCR.

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

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

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for [assignment: *list of security attributes*].

7.1.3.2 Data Protection and Privacy

FDP_RIP.1 Subset residual information protection

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

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the deallocation of the resource from[3] the following objects:
- Primary Keys,
- User keys,
- PCR data[4].

7.1.3.3 Cryptographic SFR

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

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

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

FCS_RNG.1 Random number generation

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

FCS_RNG.1.1 The TSF shall provide a *[assignment: deterministic, hybrid*[5]] random number generator that implements: NIST SP 800-90A *[assignment: Hash_DRBG, HMAC_DRBG, CTR_DRBG] [17]*[6].

---

[3] [selection: *allocation of the resource to, deallocation of the resource from*]
[4] [assignment: *list of objects*]
[5] [selection: *physical, deterministic, hybrid*]
[6] [assignment: *list of security capabilities*]

FCS_RNG.1.2 The TSF shall provide random numbers that meet: Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG[7].

Application note 2: [7], section 11.4.10, describes the RNG in the TPM as hybrid random number generator (RNG), that produces seeds by an entropy source based on physical random processes and the seeds are used for a deterministic random bit generator complying to NIST SP 800-90A [17]. NIST SP 800-90A defines the three types of deterministic random bit generators listed in the SFR and the ST author shall identify by assignment in the element FCS_RNG.1.1, which type is implemented in the TOE. The quality metric defined in the element FCS_RNG.1.2 will be fulfilled if the seeds have sufficient entropy and the assigned deterministic random number generator is correctly implemented. The Appendix 0 provides more details on evaluation of RNG. The RNG is used internally for generation of Primary Seeds, input to key generation, authorisation values and nonces.

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

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

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

Application note 3: The two selections shall be performed consistently, i.e. if RSA is selected then the key size shall be 2048 bits, if ECC is selected then the key size shall be 256 bits, if symmetric is selected then the key size shall be 128 bits and optionally 256 bits. If more than one primary key generation algorithm is supported by the TOE the ST writer shall iterate the component FCS_CKM.1/PK.

Application note 4: The ST author shall specify the key generation algorithms and key sizes that are used. The TPM library specification [7] defines two key derivation functions called KDFa and KDFe. They use a KDF in counter mode as specified in [21] with HMAC [15] as the pseudorandom function. In order to generate keys for dedicated algorithms, the values generated by the KDF may need appropriate post-processing. Examples for algorithm-specific post-processing are provided in the appendixes B and C of [7]: other methods may also be used. The ST writer shall iterate the component FCS_CKM.1 if the TOE supports more than one key generation method.

Application note 5: The EK and EPS may be generated in the manufacturing environment and injected into the TOE. The method used for injection of an EK is not addressed by this SFR.

---

[7] [assignment: *a defined quality metric*]
[8] [assignment: *cryptographic key sizes*]
[9] [assignment: list of standards]

FCS_CKM.1/ASYMM Cryptographic key generation (asymmetric keys)

Hierarchical to: No other components.

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

FCS_CKM.1.1/ASYMM The TSF shall generate cryptographic **asymmetric** keys in accordance with a specified cryptographic key generation algorithm [selection: RSA, ECC] and [assignment: other *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: TPM library specification [7], [8], [9], [assignment: *list of additional standards*].[10]

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

Hierarchical to: No other components.

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

FCS_CKM.1.1/SYMM The TSF shall generate cryptographic **symmetric** keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: TPM library specification [7], [8], [9], [assignment: *list of additional standards*].[11]

Application note 6: The refinements in the SFRs FCS_CKM.1/PK, FCS_CKM.1/ASYMM, and FCS_CKM.1/SYMM are defined in order to specify the intended usage of the generated keys more precisely. The algorithms for the generation of these cryptographic keys are dependent on the intended usage of the keys.

FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

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

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

Application note 7: FCS_CKM.4 destroys the cryptographic keys that were used by the operations as defined in FCS_COP.1. The ST author shall specify how the cryptographic keys are destroyed when not required anymore. One possible procedure may be overwriting with fixed or random data.

---

[10] [assignment: list of standards]
[11] [assignment: list of standards]

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

Hierarchical to: No other components.

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

FCS_COP.1.1/AES The TSF shall perform symmetric encryption and decryption[12] in accordance with a specified cryptographic algorithm AES in the mode CFB [selection: *CTR, OFB, CBC, and ECB*][13] and cryptographic key sizes 128, [selection: *none, 192, 256*] bits[14] that meet the following: NIST Pub 800-38a [22] or ISO/IEC 10116 [27] or ISO/IEC 18033-3 [31][15].

Application note 8: The TPM library specification [7], chapter 11.4.6, requires the TOE to implement AES in Cipher Feedback Mode (CFB) and allows support of the other block cipher modes listed for selection in the ST. ECB is not recommended. This selection may be empty. The selection of additional key sizes of AES may be empty.

FCS_COP.1/SHA Cryptographic operation (hash function)

Hierarchical to: No other components.

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

FCS_COP.1.1/SHA The TSF shall perform hash value calculation[16] in accordance with a specified cryptographic algorithm SHA-256[17] and cryptographic key sizes none[18] that meet the following: FIPS 180-4 [13][19].

Application note 9: The TPM shall implement an approved hash algorithm that has approximately the same security strength as its strongest asymmetric algorithm. If the TOE supports additional hash functions the ST writer shall iterate the component FCS_COP.1 for these hash functions.

Application note 10: The usage of the hash algorithms by the TPM shall be implemented in accordance with NIST SP 800-107 [21].

FCS_COP.1/HMAC Cryptographic operation (HMAC calculation)

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security

---

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

attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/HMAC The TSF shall perform HMAC value generation and verification[20] in accordance with a specified cryptographic algorithm HMAC and SHA-256[21] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: FIPS 198-1 [15] or ISO/IEC 9797-2 [26][22].

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

Hierarchical to: No other components.

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

FCS_COP.1.1/ASYMMED The TSF shall perform asymmetric decryption [and selection: [encryption]][23] in accordance with a specified cryptographic algorithm [selection: [RSA without padding, RSAES-PKCS1-v1_5, RSAES-OAEP], [ECDH with curve [selection: *TPM_ECC_NIST_P256*, [assignment: *other elliptic curve*][24]]][25] and cryptographic key sizes [selection: 2048 bits, 256 bits][26] that meet the following: PKCS#1v2.1 [25], TPM library specification [7], NIST Special Publication 800-56A [19] or ISO/IEC 15946-1 [30][27].

Application note 11: The selections shall be performed consistently, i.e. if RSA is selected then asymmetric encryption shall be selected and the key size shall be 2048 bits, if ECDH is selected then asymmetric encryption shall not be selected and the key size shall be 256 bits. If more than one cryptographic algorithm is supported by the TOE the ST writer shall iterate the component FCS_COP.1.1/ASYMMED.

Application note 12: The ECC key decryption does not have external command access unless implemented in the optional command TPM2_ECDH_ZGen. No ECC encryption is supported.

FCS_COP.1/Sign Cryptographic operation (signature generation/verification)

---

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

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

FCS_COP.1.1/Sign The TSF shall perform signature generation and verification[28] in accordance with a specified cryptographic algorithm [selection: RSASSA_PKCS1v1_5, RSASSA_PSS, ECDSA with curve TPM_ECC_NIST_P256 and [assignment: *other elliptic curve*]][29] and cryptographic key sizes [selection: 2048 bits, 256 bits][30] that meet the following: PKCS#1v2.1 [25] or FIPS PUB 186-4 [15] or ISO/IEC 14888-3 [29][31].

Application note 13: The two selections shall be performed consistently, i.e. if RSASSA is selected then the key size shall be 2048 bits, if ECDSA is selected then the key size shall be 256 bits. If more than one cryptographic algorithm is supported by the TOE the ST writer shall iterate the component FCS_COP.1/Sign. Signature-creation is provided by the command TPM2_Quote and optionally signature verification is provided by the recommended command TPM2_VerifySignature. The internal verify function is mandatory as defined by the SFR but access by the external command TPM2_VerifySignature is only available if implemented. The elliptic curve TPM_ECC_NIST_P256 is defined in FIPS PUB 186-4, section D.1.2.3. The ST writer shall assign any other elliptic curve supported for signature creation and verification but this assignment may be empty if no other elliptic curve is supported.

7.1.3.4 Identification and Authentication SFR

The TPM identification and authentication capability is used to authorise the use of a Protected Object and Protected Capability. Note that the TCG Library Specification document refers to the identification and authentication process and access control as *authorisation.* From the two basic mechanisms defined for authentication, only one is mandatory for this protection profile:

- the proof of knowledge of a shared secret, i.e. password or a secret for HMAC, assigned to the entity as *authValue*.

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

The *authValue* is linked to user roles. The *authValue* may be known or set to a randomly generated value. If the *authValue* is set to a randomly generated value it will be unknown to the user and the authentication is blocked.

The session based authorisation uses *handles* and random *nonces*. The handle is assigned when the session is created and identifies the session until the session is closed. The session requires that a nonce shall be used only for one message and its reply. For

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

instance, the TPM would create a nonce and send that in a reply. The requestor would receive that nonce (*nonceOlder*), generate its own nonce (*nonceNewer*) and include both values in the calculation of the command-dependent authentication value. Then, the caller sends the command, the authentication value and *nonceNewer* to the TPM which checks the authentication value with the knowledge of both nonces and executes the command on success. The nonces link commands in the command chain and commands and responses.

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

FIA_SOS.2 TSF Generation of secrets

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

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet uniform distribution of random variables generating the value.[32]

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

Application note 14: The TSF shall take the values to generate nonces from the RNG.

FMT_MSA.4/AUTH Security attribute value inheritance

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

FMT_MSA.4.1/AUTH The TSF shall use the following rules to set the value of security attributes:
(1) The bits userWithAuth in the TPMA_Object of an object are defined when the object is created and can never be changed.

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

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

FMT_MTD.1.1/AUTH The TSF shall restrict the ability to
(1) set[33] the platformAuth[34] to the role Platform firmware[35];
(2) set[36] the endorsementAuth[37] to the role Privacy Administrator[38],

---

[32] [assignment: *a defined quality metric*]
[33] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[34] [assignment: *list of TSF data*]
[35] [assignment: *the authorised identified roles*]
[36] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[37] [assignment: *list of TSF data*]
[38] [assignment: *the authorised identified roles*]

FIA_AFL.1/Recover Authentication failure handling (recovery)

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

FIA_AFL.1.1/Recover The TSF shall detect when maxTries[39] of unsuccessful authentication attempts occur related to unsuccessful password or HMAC authentication attempts for

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

FIA_AFL.1.2/Recover When the defined number of unsuccessful authentication attempts has been met[40], the TSF shall block the authorisations for RecoveryTime seconds[41].

The counter failedTries is incremented when the authentication attempt failed. The counter failedTries is decremented by one after recoveryTime seconds if:

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

Application note 15: The refinement describes the failedTries behavior. The TPM can "self-heal" after a specified amount of time.

FIA_UID.1 Timing of identification

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

FIA_UID.1.1 The TSF shall allow

(1) execution of commands that do not require authentication,
(2) access to objects where the entity owner has defined no authentication requirements (authValue),
(3) [assignment: other TSF-mediated actions][42]

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 that do not require authentication,
(2) to access objects where the entity owner has defined no authentication requirements (authValue)[43]

---

[39] [assignment: *positive integer number*], *an administrator configurable positive integer within [assignment: range of acceptable values]*]
[40] [selection: *met, surpassed*]
[41] [assignment: *list of actions*]
[42] [assignment: *list of TSF-mediated actions*]

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

FIA_UAU.5 Multiple authentication mechanisms

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

FIA_UAU.5.1 The TSF shall provide a
(1) Password based authentication mechanism,
(2) HMAC based authentication mechanism [44]
to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the **following rules:**
(1) If userWithAuth in the TPMA_Object bits is set, for operations that require USER role authorisation will be given if the caller provides proof of knowledge of the authValue of the object with an HMAC authorisation session or a password. If this attribute is CLEAR, then HMAC or password authorisations may not be used for USER role authorisations.
(2) A password based authentication mechanism is required if the authHandle parameter of the command shall contain TPM_RS_PW.
(3) A HMAC based authentication is required if the authorisation session shall be created with a sessionType of TPM_SE_HMAC[45].

Application note 17: The ST writer shall describe the implemented methods for physical presence authorisation if supported by the TOE. The Password based authentication mechanism can be used by a human user because it does not need any cryptographic calculation for authentication as required in HMAC based authentication 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 conditions that multiple commands need to be executed in one authorisation session.[46]

FIA_USB.1 User-subject binding

---

[43] [assignment: *list of TSF mediated actions*]
[44] [assignment: *list of multiple authentication mechanisms*]
[45] [assignment: *rules describing how the multiple authentication mechanisms provide authentication*]
[46] [assignment: *list of conditions under which re-authentication is required*]

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

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

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: none. [47]FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:
(1) The TSF shall create the shared secret (sessionKey) and the session handle in the case of a session based authorisation using the command TPM2_StartAuthSession.
(2) The TSF shall invalidate the shared secret (sessionKey) and the session handle in each of the following situations:
  (a) The command TPM2_FlushContext is executed for the corresponding session handle.
  (b) The flag continueSession of the session attributes is cleared.
  (c) The command TPM2_Startup is executed with the argument TPM_SU_CLEAR or TPM_SU_STATE.[48].

7.1.3.5 TSF Protection

FPT_TST.1 TSF testing
Hierarchical to: No other components.
Dependencies: No dependencies.

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

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

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

(3) [assignment: *further conditions under which self test should occur*][49]

to demonstrate the correct operation of sensitive parts of the TSF[50].

---

[47] [assignment: *rules for the initial association of attributes*]
[48] [assignment: *rules for the changing of attributes*]
[49] [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]*]
[50] [selection: *[assignment: parts of TSF], the TSF*]

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of [assignment: *parts of TSF data*][51].

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of the TSF[52].

Application note 18: The ST writer shall define additional conditions in FPT_TST.1.1 in the case that the TPM manufacturer implements additional self tests.

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

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

FPT_FLS.1.1/FS The TSF shall preserve a secure state **by entering the Fail state** when the following types of failures occur:
(1) If during TPM Restart or TPM Resume, the TPM fails to restore the state saved at the last Shutdown(STATE), the TPM shall enter Failure Mode and return TPM_RC_FAILURE,
(2) failure detected by self-test according to FPT_TST.1,
(3) [assignment: *list of additional types of failures in the TSF*][53].

Application note 19: The ST writer shall state the missing operation in the element FPT_FLS.1/FS according to the additional types of failures for which the TSF preserves a secure state if implemented by the TOE. The assignment may be "none" if no additional types of failures are handled by the TSF.

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

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

FPT_FLS.1.1/SD The TSF shall preserve a secure state **by shutdown** when the following types of failures occur:
(1) detection of a physical attack,
(2) detection of environmental condition out of spec values[54].

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*][55] to the TSF[56] by responding automatically such that the SFRs are always enforced.

---

[51] [selection: *[assignment: parts of TSF data], TSF data*]
[52] [selection: *[assignment: parts of TSF], TSF*]
[53] [assignment: *list of types of failures in the TSF*]
[54] [assignment: *list of types of failures in the TSF*]
[55] [assignment: *physical tampering scenarios*]
[56] [assignment: *list of TSF devices/elements*]

Application note 20: The ST writer shall state the missing operation in the element FPT_PHP.3 by adding specific physical tampering scenarios for which resistance is claimed for the specific TOE. This assignment may be empty.

FDP_ITT.1 Basic internal transfer protection

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

FDP_ITT.1.1 The TSF shall enforce the Access Control Policy[57] to prevent the modification and/or unauthorized access[58] to user data when it is transmitted between physically-separated parts of the TOE.

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

FPT_ITT.1 Basic internal TSF data transfer protection

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

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure[59] when it is transmitted between separate parts of the TOE.

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

7.1.4 SFRs Concerning the Object Hierarchy of the TOE

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

7.1.4.1 TPM Operational States

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

- Power-Off state: A hardware TPM is in power-off state when no power is applied to the TPM, or the power is on and a reset is being asserted. This state may be reached from any other state because power can be lost at any time. In that sense, Figure 3 is incomplete because some possible state transitions are not shown for clarity

[57] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[58] [assignment: *disclosure, modification, loss of use*]
[59] [assignment: *disclosure, modification, loss of use*]

reasons. The TPM does not execute any function except transition to the Init state when it receives the _TPM_Init indication.

- Initialisation state: The TPM enters this state when it receives the _TPM_Init indication. This indication is provided in a platform-specific manner (cf. section 12.2.2 of [7] for details). In the Init state, only the commands TPM2_Startup is accepted. All other commands do not change the state and cause an error return code. The TPM may perform self-test in the Init state and may enter Failure mode if the self-test detects any failure.
- Operational state: In this state the TPM was successfully initialised. The initialisation of the operational status of the TPM is done by the TPM2_Startup command and may restore a previously (by TPM2_Shutdown) saved status. Details are defined in section 12.2.3 and 12.2.4 of [7]. In the Operational state, there are no restrictions on the set of commands that are accepted. Before the TPM may return a result based on a cryptographic algorithm, it is required to perform a specific self-test of that algorithm. If a command requires use of an untested algorithm or functional module, the TPM performs the test and then completes the command actions. This behavior is modeled in Figure 3 using a state transition to Self-Test.

  Note that the TPM2_Shutdown command does not imply a reset or any state change of the TPM: It is used to prepare the TPM for a power cycle and may be used to save the operational status of the TPM for a later restore. Details can be found in section 11.4 of [9].
- Self-Test state: This state implements the required tests of cryptographic algorithms and is not triggered by a dedicated TPM command. When performing a self-test on demand, the TPM should test only those algorithms needed to complete the command. The command TPM2_SelfTest may optionally cause the TPM to trigger a full self-test of all algorithms and functional blocks. Depending on the result, the TPM changes its state back to Operational or to Fail after completion of the self-test.
- Fail state: In Fail state the TPM does not allow any command except TPM2_GetTestResult and TPM2_GetCapability. The only way the TPM exits the Fail state is when it receives the _TPM_Init indication.

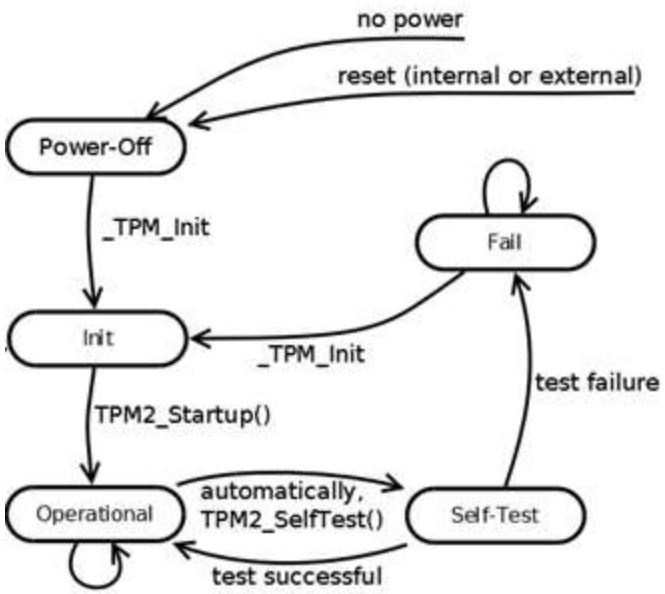


Figure 3: States of the TPM and its Transitions (informative)

Application note 21: Figure 3 illustrates the transitions between the TPM operational states as defined in the library specification, chapter 12 of [7].

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

Table 9: objects, operations and security attributes for the TPM state control SFP

#	Protected Objects	Operations	Security attributes
1	**Shutdown BLOB** A set of variables that represent the operational status of the TPM as it is in the Operational state (see Figure 3).	**Generate** The shutdown BLOB is written to the NV memory by the command TPM2_Shutdown with parameter TPM_SU_STATE. **Resume** The shutdown BLOB is read from the NV memory by the command TPM2_Startup with parameter TPM_SU_STATE. The operational variables are restored with the values from the shutdown BLOB. This is called “TPM RESUME”, see section 11.3 in [9]. **Restart** The shutdown BLOB is read	Security attributes: **Validation status**, used to check the validity of the Shutdown BLOB. After Generation of the Shutdown BLOB its validation status is positive. The execution of some commands may invalidate this status. The conditions that invalidate this validation status are defined in section 12.2.4 of [7]. In that document the BLOB is called “saved TPM state”.

#	Protected Objects	Operations	Security attributes
		from the NV memory by the command TPM2_Startup with the parameter TPM_SU_CLEAR. Some operational variables are restored with the values from the shutdown BLOB. This is called "TPM RESTART", see section 11.3 in [9].	

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

Hierarchical to: FDP_ACC.1 Subset access control
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1/States The TSF shall enforce the TPM State Control SFP[60] on all subjects and objects[61] and all operations among subjects and objects covered by the SFP.

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

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

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

FDP_ACF.1.1/States The TSF shall enforce the TPM State Control SFP[62] to objects based on the following
Subjects as defined in Table 7:
(1) Platform firmware with the security attributes platformAuth and physical presence if supported by the TOE,
(2) all other subjects; their security attributes are irrelevant for this SFP,
Objects as defined in Table 8 and Table 9:
(1) Shutdown BLOB with the security attribute validation status,
(2) all other objects; their security attributes are irrelevant for this SFP[63].

FDP_ACF.1.2/States The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
(1) In the Init state the subject "World" is authorised to execute the commands TPM2_Startup, _TPM2_FieldUpgradeStart, and TPM2_FieldUpgradeData.

---

[60] [assignment: *access control SFP*]
[61] [assignment: *list of subjects and objects*]
[62] [assignment: *access control SFP*]
[63] [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) In the Init state every subject is authorised to process the Resume operation on the Shutdown BLOB with state transition to Operational.
(3) In the Init state every subject is authorised to process the Restart operation on the Shutdown BLOB with state transition to Operational.
(4) In the Init state, if no Shutdown BLOB was generated or if the Shutdown BLOB is invalid (see attribute "Validation status") every subject is authorised to process the TPM2_Startup command. In the case of the parameter TPM_SU_CLEAR the TPM shall change the state to Operational and initialise its internal operational variables to default initialisation values (Reset), otherwise the TPM shall return TPM_RC_FAILURE and stay in the same state.
(5) In the Operational state, nobody is authorised to execute the command TPM2_Startup. For all other subjects, objects and operations, the access control rules of the Access Control SFP shall apply (see FDP_ACF.1/AC).
(6) The Operational state shall change to Self-Test state if the command TPM2_Selftest is executed or when a test of dedicated functionality is required (see FPT_TST.1). In the Self-Test state, nobody is authorised to execute any other TPM command.
(7) The Self-Test state shall be left only after finishing the intended test of the dedicated functionality. In the case of a successful test result the state shall change to Operational, otherwise to Fail.
(8) In the Fail state, every subject is authorised to execute the commands TPM2_GetTestResult and TPM2_GetCapability.
(9) In the Fail state the subject World is authorised to send a _TPM_Init indication with state change to Init.
(10) Any subject is authorised to prepare the TPM for a power cycle using the TPM2_Shutdown command and to create a shutdown BLOB by TPM2_Shutdown(TPM_SU_STATE).[64]

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

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

(1) Once the TPM receives a TPM2_SelfTest command and before completion of all tests, the TPM shall return TPM_RC_TESTING for any command that uses a function that requires a test.[65]

Application note 22: The _TPM_Init indication is normally signaled by the de-assertion of the TPM's reset signal. It may also be signaled by an interface protocol or setting.

[64] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[65] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

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

Hierarchical to: No other components.

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

FMT_MSA.1.1/States TSF shall enforce the TPM state control SFP[66] to restrict the ability to modify[67] the security attributes TPM state:
(1) to any role[68].

Application note 23: The concrete restrictions in the TPM state control SFP to restrict the modification of the TPM state by dedicated roles is defined in FMT_MSA.1/States.

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

Hierarchical to: No other components.

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

FMT_MSA.3.1/States The TSF shall enforce the TPM state control SFP[69] to provide restrictive[70] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/States The TSF shall allow ~~the~~ nobody[71] to specify alternative initial values to override the default values when an object or information is created.

7.1.4.2 Creation and Modification of the TPM Hierarchy

Hierarchies are characterised by a parent-child relationship of objects. The TPM supports 2 hierarchies: the platform hierarchy, and the endorsement hierarchy and may optionally support the storage hierarchy. For (temporary) objects that are used only until the next TPM reset, a temporary object hierarchy may be created, if supported by the TPM. The root of each TPM hierarchy is defined by a primary seed: Primary seeds are random values that are persistently stored in a TPM. The children of primary seeds are called primary objects.

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

If the hierarchy is disabled the authValue is not applicable.

FDP_SDI.1 Stored data integrity monitoring

---

66 [assignment: *access control SFP(s), information flow control SFP(s)*]
67 [selection: *change_default, query, modify, delete, [assignment: other operations]*]
68 [assignment: *the authorised identified roles*]
69 [assignment: *access control SFP, information flow control SFP*]
70 [selection, choose one of*: restrictive, permissive, [assignment: other property]*]
71 [assignment: *the authorised identified roles*]

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

FDP_SDI.1.1 The TSF shall monitor user data stored in containers controlled by the TSF for data modifications and modification of hierarchy[72] on all objects, based on the following attributes: HMAC over the sensitive area of an object of the TPM hierarchy, object creation ticket[73].

Application note 24: The attributes mentioned in FDP_SDI.1 shall be generated at object creation time using the command TPM2_Create or TPM2_CreatePrimary. The HMAC over the sensitive data shall be done according to the section 27.7 of [7]. The object creation ticket (see section 10.7.3 of [8]) proves the environment in the object hierarchy at object creation time.

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

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

FDP_ACC.1.1/Hier The TSF shall enforce the TPM Object Hierarchy SFP[74] on

Subjects
(1) Platform firmware,
(2) Privacy administrator,
(3) USER,
(4) World

Objects
(1) PPS,
(2) EPS,
(3) PPO,
(4) EK,
(5) object in a TPM hierarchy

Operations
(1) TPM2_CreatePrimary,
(2) TPM2_HierarchyChangeAuth,
(3) TPM2_Load,
(4) TPM2_ReadPublic,
(5) Use.[75]

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

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

FDP_ACF.1.1/Hier The TSF shall enforce the TPM Object Hierarchy SFP[76] to objects based on the following:

---

[72] [assignment: *integrity errors*]
[73] [assignment: *user data attributes*]
[74] [assignment: *access control SFP*]
[75] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

Subjects:
(1) Platform firmware with security attribute authorisation state gained by authentication with platformAuth,
(2) Privacy administrator with security attribute authorisation state gained by authentication with endorsementAuth,
(3) USER with authentication state gained with authValue,
(4) World with no security attributes,

Objects:
(1) EPS,
(2) PPS,
(3) EK,
(4) PPO,
(5) object in a TPM hierarchy with security attributes: state of the hierarchy, fixedParent, fixedTpm[77]

FDP_ACF.1.2/Hier The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
(1) The subject World is authorised to create an EPS whenever the TPM is powered on and no EPS is present.
(2) The subject World is authorised to create an PPS whenever the TPM is powered on and no PPS is present.
(3) The Platform firmware is authorised to create a Platform Primary Object under PPS. The physical presence is not required if it is not supported by the TOE or disabled for TPM2_CreatePrimary command.
(4) The privacy administrator is authorised to create a primary object (EK) under EPS.
(5) The Platform firmware with platformAuth or physical presence if supported by the TOE, and the privacy administrator are authorised to change the authorisation secret for a hierarchy (TPM2_HierarchyChangeAuth). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_HierarchyChangeAuth command.[78]

FDP_ACF.1.3/Hier The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none[79].

FDP_ACF.1.4/Hier The TSF shall explicitly deny access of subjects to objects based on the following additional rules:
(1) none[80].

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

---

[76] [assignment: *access control SFP*]
[77] [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*]
[78] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[79] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[80] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

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

FMT_MSA.1.1/Hier TSF shall enforce the TPM Object Hierarchy SFP[81] to restrict the ability to modify[82] the security attributes fixedTPM and fixedParent[83] to nobody[84].

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

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

FMT_MSA.3.1/Hier The TSF shall enforce the TPM Object Hierarchy SFP[85] to provide restrictive[86] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Hier The TSF shall allow the creator of an object in a TPM hierarchy[87] to specify alternative initial values to override the default values when an object or information is created.

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

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

FMT_MSA.4.1/Hier The TSF shall use the following rules to set the value of security attributes: none[88]

7.1.4.3 Data Import and Export

The TPM supports the creation of hierarchies of entities. A hierarchy is constructed with Storage Keys as the connectors to which other types of objects may be attached.

In order to summarise the correlations of different TPM commands regarding data import and export, Figure 4 illustrates possible scenarios: To be able to use an object as part of the TPM hierarchy, it needs to be already loaded. The load operation is implemented as TPM2_Load command. The TPM2_Load command requires a TPM object that could have been created by TPM2_Create from an object template.

---

[81] [assignment: *access control SFP(s), information flow control SFP(s)*]
[82] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[83] [assignment: *list of security attributes*]
[84] [assignment: *the authorised identified roles*]
[85] [assignment: *access control SFP, information flow control SFP*]
[86] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[87] [assignment: *the authorised identified roles*]
[88] [assignment: *rules for setting the values of security attributes*]

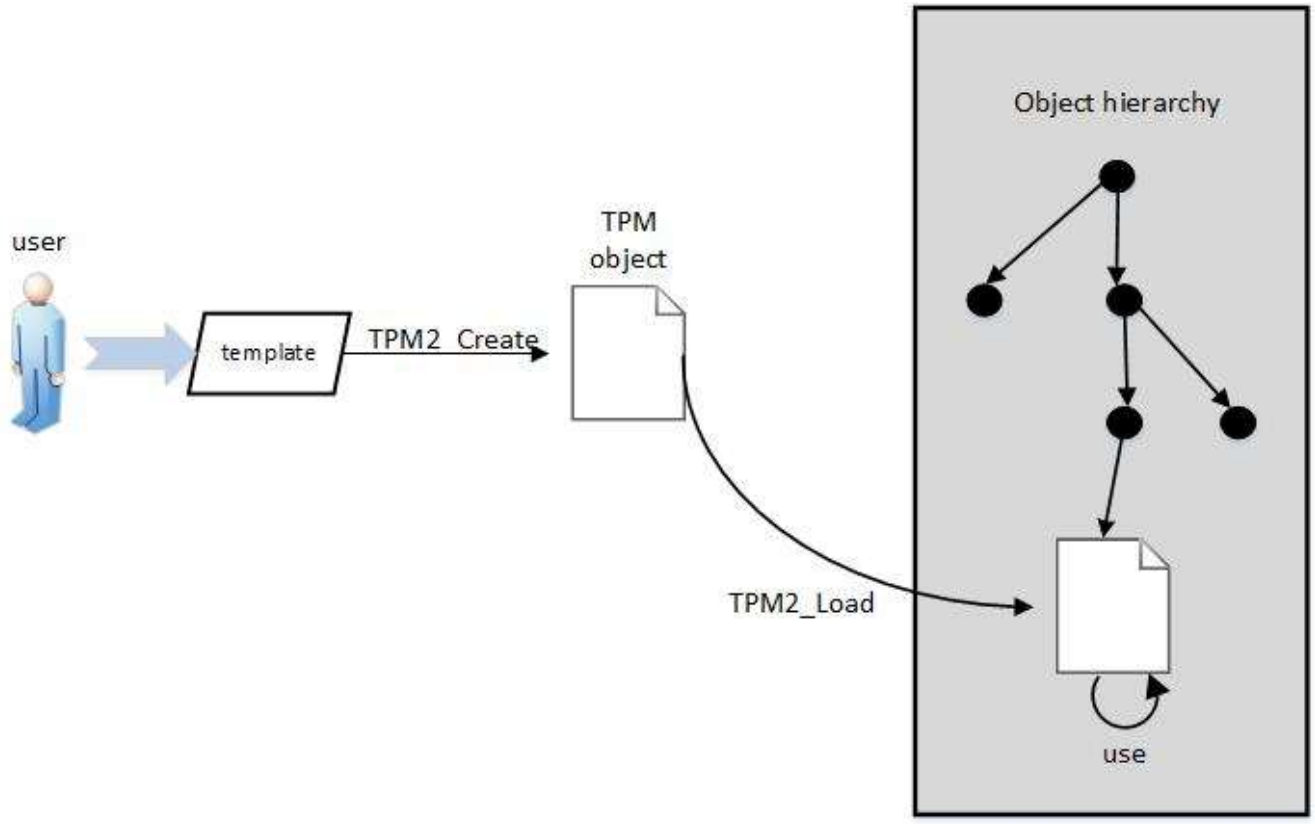


Figure 4: Object Export/Import Scenarios (informative)

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

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

FDP_ACC.1.1/ExIm The TSF shall enforce the Data Export and Import SFP[89] on
Subjects:
(1) USER,
(2) World
Objects:
(1) Platform Primary Object,
(2) Endorsement Primary Key,
(3) User Key,

Operations:
(1) export by means of TPM2_Create,
(2) load by means of TPM2_Load. [90]
(3) import by means of TPM2_Import,

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

[89] [assignment: *access control SFP*]
[90] [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*]

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

FDP_ACF.1.1/ExIm The TSF shall enforce the Data Export and Import SFP[91] to objects based on the following:
Subjects:
(1) USER with authentication state gained with authValue,
(2) World without any successful authentication
Objects:
(1) Platform Primary Object with the security attributes platformAuth,
(2) Endorsement Primary Key with the security attributes authorisation data
(3) User Key with the security attributes authorisation data. [92]

FDP_ACF.1.2/ExIm The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
(1) The subject USER is authorised to export an object using the TPM2_Create command.
(2) The subject USER authorised for the parent object is allowed to load objects into the TPM hierarchy using the command TPM2_Load.
(3) The subject USER authorised for the parent object is allowed to import an object using the TPM2_Import command under the following conditions:
    (a) The attributes “fixedTPM” and “fixedParent” of the object shall not be set.
    (b) If an encryption of the object to import is performed, then an integrity evidence value shall be part of the imported object.
    (c) If an integrity evidence value is present, the object shall only be imported after the integrity was successfully verified.
(4) The subject World is authorised to read the public portion of a TPM object using the command TPM2_ReadPublic. [93]

FDP_ACF.1.3/ExIm The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none[94]

FDP_ACF.1.4/ExIm The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[95].

---

[91] [assignment: *access control SFP*]
[92] [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*]
[93] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[94] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[95] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

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

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

FMT_MSA.1.1/ExIm TSF shall enforce the Data Export and Import SFP[96] to restrict the ability to use[97] the security attributes authorisation data[98] to every subject[99].

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

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

FMT_MSA.3.1/ExIm The TSF shall enforce the Data Export and Import SFP[100] to provide restrictive[101] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/ExIm The TSF shall allow ~~the~~ nobody[102] to specify alternative initial values to override the default values when an object or information is created.

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

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

FDP_ETC.2.1/ExIm The TSF shall enforce the Data Export and Import SFP[103] when exporting user data, controlled under the SFP(s), outside of the TOE.

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

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

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

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

---

[96] [assignment: *access control SFP(s), information flow control SFP(s)*]
[97] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[98] [assignment: *list of security attributes*]
[99] [assignment: *the authorised identified roles*]
[100] [assignment: *access control SFP, information flow control SFP*]
[101] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[102] [assignment: *the authorised identified roles*]
[103] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

(2) The sensitive area of an object from the TPM hierarchy shall be symmetrically encrypted before its export using the command TPM2_Create. The used key and the IV should be derived from the secret seed of the parent in the TPM hierarchy. [104]

Application note 25: The details of the derivation of the key and IV for the symmetric encryption and HMAC generation for export of the sensitive area of objects are specified in section 22.4 and 22.5 of [7].

.

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

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

FDP_ITC.2.1/ExIm The TSF shall enforce the Data Export and Import SFP[105] when importing user data, controlled under the SFP, from outside of the TOE.

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

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

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

FDP_ITC.2.5/ExIm The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:
(1) If an inner or an outer wrapper is present then a valid integrity value shall be present.[106]

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

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

FDP_UCT.1.1/ExIm The TSF shall enforce the Data Export and Import SFP[107] to transmit[108] user data in a manner protected from unauthorised disclosure.

---

[104] [assignment: *additional exportation control rules*]
[105] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[106] [assignment: *additional importation control rules*]

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

Hierarchical to: No other components.

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

FDP_UIT.1.1/ExIm The TSF shall enforce the Data Export and Import SFP[109] to transmit and receive[110] user data in a manner protected from modification[111] errors.

FDP_UIT.1.2/ExIm The TSF shall be able to determine on receipt of user data, whether modification[112] has occurred.

7.1.4.4 Measurement and Reporting

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

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

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/M&R The TSF shall enforce the Measurement and Reporting SFP[113] on

subjects
(1) Platform firmware,
(2) USER,
(3) World,

objects
(1) PCR,
(2) TPM objects,

operations
(1) TPM2_PCR_Extend,
(2) TPM2_PCR_Event,
(3) TPM2_PCR_Read,
(4) TPM2_Quote[114]

---

[107] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[108] [selection: *transmit, receive*]
[109] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[110] [selection: *transmit, receive*]
[111] [selection: *modification, deletion, insertion, replay*]
[112] [selection: *modification, deletion, insertion, replay*]
[113] [assignment: *access control SFP*]
[114] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

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

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

FDP_ACF.1.1/M&R The TSF shall enforce the Measurement and Reporting SFP[115] to objects based on the following:
Subjects:
(1) Platform firmware with security attribute authorisation state gained by authentication with platformAuth,
(2) USER with authentication state gained with authValue,
(3) World with no security attributes,
Objects:
(1) PCR with the security attribute PCR-attributes TPM_PT_PCR,
(2) TPM objects with the security attributes authentication data (authValue)[116]

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) Authorised subjects of role USER are allowed to extend the PCR using the command TPM2_PCR_Extend.
(2) Authorized subjects of role USER are allowed to update the PCR using the command TPM2_PCR_Event .
(3) The subject World is authorised to read values of PCR using the command TPM2_PCR_Read.
(4) Authorised subjects of role USER are allowed to quote PCR values using the command TPM2_Quote. The authorisation shall be done based on the key that is used for the quotation.[117]

FDP_ACF.1.3/M&R The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none[118].

FDP_ACF.1.4/M&R The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none[119].

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

---

[115] [assignment: *access control SFP*]
[116] [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*]
[117] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[118] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[119] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

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

FMT_MSA.1.1/M&R TSF shall enforce the Measurement and Reporting SFP[120] to restrict the ability to query[121] the security attributes PCR attributes[122] to World[123].

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

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

FMT_MSA.3.1/M&R The TSF shall enforce the Measurement and Reporting SFP[124] to provide restrictive[125] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/M&R The TSF shall allow ~~the~~ nobody[126] to specify alternative initial values to override the default values when an object or information is created.

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

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

FCO_NRO.1.1/M&R The TSF shall be able to generate evidence of origin for transmitted attestation structure (TPM2B_ATTEST) and object creation tickets[127] at the request of the originator[128].

FCO_NRO.1.2/M&R The TSF shall be able to relate the
(1) magic number for an identification of whether the TPM produced the signed digest,
(2) type of the attestation structure indicating the contents of the attested parameter,
(3) qualified name of the key used to sign the attestation data (qualifiedSigner),
(4) external information supplied by the caller,
(5) the firmware version[129]

of the originator of the information, and the command dependent value of the
(1) PCR data (using the command TPM2_Quote)[130]

---

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

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[131] given as soon as the recipient can verify the signature and has confidence in the key that is used to sign[132].

Application note 26: The key used for signing may be any key with the sign attribute set. If a key is not restricted to a dedicated scheme then the caller of the corresponding command may indicate the signing scheme to be used.

7.1.5 SFRs for the TOE Operation

7.1.5.1 Access SFR

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

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/AC The TSF shall enforce the Access Control SFP*[133]* on

subjects

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

objects

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

operations

(1) TPM2_EvictControl,
(2) TPM2_HashSequenceStart,
(3) TPM2_SequenceUpdate,
(4) TPM2_EventSequenceComplete,
(5) TPM2_Hash[134].

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

---

[130] [assignment: *list of information fields*]
[131] [selection: *originator, recipient, [assignment: list of third parties]*]
[132] [assignment: *limitations on the evidence of origin*]
[133] [assignment: *access control SFP*]
[134] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

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

FDP_ACF.1.1/AC The TSF shall enforce the Access Control SFP[135] to objects based on the following
Subjects:
(1) Platform firmware with security attribute authorisation state gained by authentication with platformAuth or physical presence if supported by the TOE,
(2) Privacy administrator with security attribute authorisation state gained by authentication with endorsementAuth,
(3) USER with authentication state gained with authValue,
(4) World with no security attributes,
Objects:
(1) User key with security attributes TPM_ALG_ID, TPMA_OBJECT,
(2) TPM objects,
(3) Data with security attribute "externally provided"[136].

FDP_ACF.1.2/AC The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
(1) The Platform firmware authorized with platformAuth, or with physical presence if supported by the TOE, is authorised to control the persistence of loadable objects in TPM memory (TPM2_EvictControl). The physical presence is not required if it is not supported by the TOE or disabled for TPM2_EvictControl command.
(2) The subject World is authorised to start a hash or event sequence using the command TPM2_HashSequenceStart.
(3) The subject USER is authorised to add data to a hash, event or HMAC sequence using the command TPM2_SequenceUpdate.
(4) The subject USER is authorised to add the last part of data (if any) to a hash or HMAC sequence using the command TPM2_ SequenceComplete.
(5) The subject USER is authorised to add the last part of data (if any) to an event sequence using the command TPM2_EventSequenceComplete.
(6) Any subject is authorised to perform hash operations on a data buffer using the command TPM2_Hash. [137]

FDP_ACF.1.3/AC The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*][138]

---

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

FDP_ACF.1.4/AC The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].[139]

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

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

FMT_MSA.1.1/AC TSF shall enforce the Access Control SFP[140] to restrict the ability to interact[141] with any attribute[142] to none[143]

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

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

FMT_MSA.3.1/AC The TSF shall enforce the Access Control SFP[144] to p**r**ovide restrictive[145] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/AC The TSF shall allow the USER[146] to specify alternative initial values to override the default values when an object or information is created.

Application note 27: The default values are defined on object creation using the command TPM2_Create or TPM2_CreatePrimary.

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

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

FDP_UCT.1.1/AC The TSF shall enforce the Access Control SFP[147] to transmit[148] user data in a manner protected from unauthorised disclosure.

---

[139] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[140] [assignment: *access control SFP(s), information flow control SFP(s)*]
[141] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[142] [assignment: *list of security attributes*]
[143] [assignment: *the authorised identified roles*]
[144] [assignment: *access control SFP, information flow control SFP*]
[145] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[146] [assignment: *the authorised identified roles*]
[147] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[148] [selection: *transmit, receive*]

Application note 28: The SFR FDP_UCT.1/AC requires the ability to encrypt the command data in a TPM command.

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

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

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

FTP_ITC.1.2 The TSF shall permit another trusted IT product[149] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for
(1) an authorisation session,
(2) an encryption session, identified by the encrypt or decrypt attribute of the session
in order to transfer commands and responses between the other trusted IT product and the TOE.[150]

Application note 29: An authorisation session or an encryption session is established by the command TPM2_StartAuthSession. The integrity protection of an authorisation session shall be implemented using a HMAC digest over the command or response data including the parameters as defined in [7]. In an encrypted session, only the first parameter shall be encrypted as long as the parameter has a size field [7].

7.1.5.2 Non-Volatile Storage

The non-volatile memory (NV memory) is used to retain values across power events. In particular, the following values are stored in the NV memory:

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

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

---

[149] [selection: *the TSF, another trusted IT product*]
[150] [assignment: *list of functions for which a trusted channel is required*]

Application note 30: The TPM library specification allows usage of an external device for storing non-volatile NV data (see section 37.7.2 of [7]). If this option will be implemented, the ST writer shall model this inter-TSF user data transfer by additional SFRs FDP_UCT.1 and FDP_UIT.1.

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

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

FDP_ACC.1.1/NVM The TSF shall enforce the NVM SFP[151] on

Subjects:
(1) Platform firmware,
(2) USER,
(3) World

Objects:
(1) ordinary NV index,
(2) objects of the TPM hierarchy

Operations:
(1) TPM2_NV_Read
(2) TPM2_EvictControl[152].

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

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

FDP_ACF.1.1/NVM The TSF shall enforce the NVM SFP[153] to objects based on the following:

Subjects as defined in Table 7:
(1) Platform firmware, USER, World with the security attributes
  (a) authentication status,
  (b) physical presence if supported by the TOE

Objects as defined in Table 8:
(1) NV index with the security attributes:
  (a) NV attributes,
  (b) status whether physical presence is required for Platform firmware authorisation[154]

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

(1) The subject platform firmware with platformAuth or physical presence if supported by the TOE, are authorised to import transient TPM objects if they are part of any TPM hierarchy, if the object attributes allow the import and if the objects contain both public and private portions. This shall be

[151] [assignment: *access control SFP*]
[152] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[153] [assignment: *access control SFP*]
[154] [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*]

done by the command TPM2_EvictControl. Physical presence is not required if it is not supported by the TOE or disabled for the TPM2_EvictControl command.

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

(3) The subject Platform firmware with the role USER is authorized to read a NV Index by the command TPM2_NV_Read if the TPMA_NV_PPREAD value of the NV index attribute is set and the NV index is not temporarily blocked by its attribute TPMA_NV_READ_READLOCKED. If the TPMA_NV_AUTHREAD attribute is set then authentication shall use authValue of the index.[155]

FDP_ACF.1.3/NVM The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none[156].

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

(1) If phEnableNV is CLEAR

a) NV indices that have TPMA_PLATFORM_CREATE SET may not be read by TPM2_NV_Read

b) The platform cannot define (TPM_RC_HIERARCHY) or TPM_RC_HANDLE) indices[157].

Application note 31: The blocking of read access to NV indices shall be reset on TPM Reset or TPM Restart. This is addressed in the TPM state control SFP, see FDP_ACF.1/States and Table 8.

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

Hierarchical to: No other components.

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

FMT_MSA.3.1/NVM The TSF shall enforce the NVM SFP[158] to provide restrictive[159] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/NVM The TSF shall allow ~~the~~ nobody[160] to specify alternative initial values to override the default values when an object or information is created.

Application note 32: The NV Index defined for the Endorsement Key Credential may be defined with attributes selected by the index creator.

---

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

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

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

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

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

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

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

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

FMT_MSA.4.1/NVM The TSF shall use the following rules to set the value of security attributes: none.[161]

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

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

FDP_ITC.1.1/NVM The TSF shall enforce the NVM SFP[162] when importing user data, controlled under the SFP, from outside of the TOE.

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

FDP_ITC.1.3/NVM The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: none[163]

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

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

FDP_ETC.1.1/NVM The TSF shall enforce the NVM SFP[164] when exporting user data, controlled under the SFP(s), outside of the TOE.

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

7.2 Security assurance requirements

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

Table 10: Security assurance requirements for the TOE

161

162 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
163 [assignment: *additional importation control rules*]
164 [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

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.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.4 Methodical vulnerability analysis

7.3 Security Requirements rationale

7.3.1 Sufficiency of SFR

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

Table 11: Security requirements rationale

	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Leak_Resistance
FMT_SMR.1														x				
FMT_SMF.1													x					
FMT_MSA.2													x					
FCS_RNG.1	x																	
FIA_SOS.2						x												
FMT_MTD.1/AUTH						x												
FIA_AFL.1/Recover						x										x		
FIA_UID.1						x		x										
FIA_UAU.1						x		x										
FIA_UAU.5						x			x								x	
FIA_UAU.6						x										x		
FIA_USB.1		x				x								x				
FPT_TST.1				x	x										x			
FMT_MSA.4/AUTH		x				x							x					
FDP_ACC.2/States		x																
FDP_ACF.1/States		x																
FMT_MSA.1/States		x											x					
FMT_MSA.3/States		x											x					
FDP_ACC.1/AC		x																
FDP_ACF.1/AC		x																
FMT_MSA.1/AC		x											x					
FMT_MSA.3/AC		x											x					
FDP_UCT.1/AC																	x	

	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.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Leak_Resistance
FTP_ITC.1/AC																	x	
FCS_CKM.1/PK	x																	
FCS_CKM.1/ASYMM	x																	
FCS_CKM.1/SYMM	x																	
FCS_CKM.4	x																	
FCS_COP.1/AES	x		x				x										x	
FCS_COP.1/SHA									x	x							x	
FCS_COP.1/HMAC	x		x			x	x										x	
FCS_COP.1/ASYMMED			x				x										x	
FCS_COP.1/Sign										x		x						
FDP_ACC.1/NVM		x																
FDP_ACF.1/NVM		x																
FMT_MSA.3/NVM		x											x					
FMT_MSA.4/NVM		x											x					
FDP_ITC.1/NVM							x											
FDP_ETC.1/NVM			x															
FDP_ACC.1/ExIm		x	x				x											
FDP_ACF.1/ExIm		x	x				x											
FMT_MSA.1/ExIm		x	x				x						x					
FMT_MSA.3/ExIm		x	x				x						x					
FDP_ETC.2/ExIm			x															
FDP_ITC.2/ExIm							x											
FDP_UCT.1/ExIm			x				x											
FDP_UIT.1/ExIm			x				x			x								
FDP_ACC.1/M&R		x							x									
FDP_ACF.1/M&R		x							x									
FMT_MSA.1/M&R		x							x			x	x					
FMT_MSA.3/M&R		x							x			x	x					
FCO_NRO.1/M&R										x		x						
FDP_RIP.1											x							

	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.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Leak_Resistance
FPT_FLS.1/FS				x											x			
FPT_FLS.1/SD				x											x			
FPT_PHP.3																		x
FDP_ITT.1																		x
FPT_ITT.1																		x
FDP_SDI.1			x		x		x											
FDP_ACC.1/Hier		x																
FDP_ACF.1/Hier		x																
FMT_MSA.1/Hier		x											x					
FMT_MSA.3/Hier		x											x					
FMT_MSA.4/Hier		x											x					

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

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

- FCS_CKM.1/PK requires the TSF to generate primary cryptographic keys by means of defined key generation functions.
- FCS_CKM.1/ASYMM requires the TSF to generate asymmetric cryptographic keys in accordance with an assigned key generation algorithm.
- FCS_CKM.1/SYMM requires the TSF to generate cryptographic symmetric keys in accordance with an assigned key generation algorithm.
- FCS_CKM.4 requires the TSF to be able destroy cryptographic keys in accordance with a specific key destruction method.
- FCS_COP.1/HMAC requires that the TSF provides the ability to generate and verify HMAC values.
- FCS_COP.1/AES requires that the TSF provides the ability to perform symmetric encryption and decryption of data.

- FCS_RNG.1 requires the TSF to provide a random number generator. This is used as a source of randomness in the specified key generation algorithm according to FCS_CKM.1/PK, FCS_CKM.1/ASYMM, and FCS_CKM.1/SYMM.

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

- FIA_USB.1 addresses the associations between subjects and its security attributes. The SFR defines rules for initial association and changes of these associations.
- FDP_ACC.2/States requires that the TSF enforces the TPM State Control SFP on all subjects, objects and operations among subjects and objects covered by the SFP. The operations shall be covered by an access control SFP.
- FDP_ACF.1/States defines rules to enforce a policy regarding the TOE states, transitions between states, and required authorisations to change the state of the TOE.
- FMT_MSA.1/States requires that a TSF shall enforce the TPM State Control SFP to restrict the ability to modify the TOE state.
- FMT_MSA.3/States requires that the TSF shall enforce the TPM State Control SFP to provide restrictive default values for security attributes and nobody is authorised to specify alternative default initial values.
- FDP_ACC.1/AC requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects, and operations among subjects and objects covered by the SFP.
- FDP_ACF.1/AC defines rules to enforce a policy regarding the TOE access control and the required authorisations to perform dedicated operations.
- FMT_MSA.1/AC requires that a TSF shall enforce a SFP to restrict the ability to perform dedicated operations on security attributes to dedicated authorised roles.
- FMT_MSA.3/AC requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes and only dedicated roles are authorised to specify alternative default initial values.
- FMT_MSA.4/AUTH defines rules to configure the object access control with authValue.
- FDP_ACC.1/NVM requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects, and NVM related operations among subjects and objects covered by the SFP.
- FDP_ACF.1/NVM defines rules to enforce a policy regarding the NVM access control and the required authorisations to perform dedicated operations.
- FMT_MSA.3/NVM requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes and nobody is authorised to specify alternative default initial values.
- FMT_MSA.4/NVM requires management of security attributes controlling read access to NVM.
- FDP_ACC.1/ExIm requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects, and export/import operations among subjects and objects covered by the SFP.

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

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

- FCS_COP.1/ASYMMED requires that the TSF provides the ability to perform asymmetric encryption and decryption of data.
- FCS_COP.1/AES requires that the TSF provides the ability to perform symmetric encryption and decryption of data.
- FCS_COP.1/HMAC requires that the TSF provides the ability to generate and verify HMAC values.
- FDP_ETC.1/NVM requires that the TSF enforces a SFP when exporting user data from NV memory.
- FDP_ACC.1/ExIm requires that the TSF enforces a SPF for access control regarding dedicated subjects, objects, and export/import operations among subjects and objects covered by the SFP.

- FDP_ACF.1/ExIm defines rules to enforce a policy regarding the access control and the required authorisations to perform export and import related operations.
- FMT_MSA.1/ExIm requires that a TSF shall enforce a SFP to restrict the ability to use the security attribute *authorisation data* for export and import related functions.
- FMT_MSA.3/ExIm requires that the TSF shall enforce a SFP to provide restrictive default values for security attributes for export and import related functions. Nobody is authorised to specify alternative default initial values for those security attributes.
- FDP_ETC.2/ExIm requires that the TSF shall apply a policy when exporting user data. The policy rules require the protection of data integrity and confidentiality at export of objects from the TPM hierarchy.
- FDP_UCT.1/ExIm requires that the TSF protects the confidentiality of transmitted user data during data export and import.
- FDP_UIT.1/ExIm requires that the TSF protects the integrity of transmitted user data during data export and import.
- FDP_SDI.1 requires that the TSF shall monitor stored user data for modification and modification of the TPM hierarchies.

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

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

- FPT_TST.1 requires the TSF to provide self tests in order to detect failure situations. These self tests may include tests of the system and data integrity.
- FDP_SDI.1 requires that the TSF shall monitor stored user data for modification and modification of the TPM hierarchies.

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

- FIA_SOS.2 requires the TSF to generate secrets for usage in the authentication functionality.
- FMT_MTD.1/AUTH requires the TSF to restrict the management of authentication data to dedicated authorised roles.
- FMT_MSA.4/AUTH defines rules for management of the security attributes controlling the use of authentication mechanisms for authorisation of objects.
- FIA_AFL.1/Recover requires the TSF to detect attacks to the authentication system by a number of ongoing unsuccessful authentication requests. On detection the TSF shall block that authentication method.
- FIA_UID.1 requires the TSF to allow dedicated commands before a user is identified. For any other TSF mediated action the TSF shall require the successful identification of the user.

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

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

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

The security objective **O.Limit_Actions_Auth** requires that the TOE restricts the actions a user may perform before the TOE verified the identity of the user. This includes requirements for physical presence provided by a user as verified by the platform firmware if physical presence is supported and enabled for the required command. This is directly addressed by the following SFRs:

- FIA_UAU.1 which requires the TSF to allow only dedicated commands before a user is authenticated. For any other TSF mediated action the TSF shall require the successful authentication of the user.
- FIA_UID.1 requires the TSF to allow dedicated commands before a user is identified. For any other TSF mediated action the TSF shall require the successful identification of the user.

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

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

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

- FCS_COP.1/SHA requires the TSF to be able to perform hash value calculations. This can be used to support data integrity protection.
- FCS_COP.1/Sign requires the TSF to be able to perform signature generation and verification. This can be used to support source authentication and source non-repudiation when exchanging data with a remote system.
- FDP_UIT.1/ExIm requires that the TSF protects the integrity of transmitted user data during data export and import.
- FCO_NRO.1/M&R requires the TSF to be able to generate evidence of origin for transmitted attestation structures and to verify this evidence.

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

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

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

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

- FMT_SMF.1 requires the TSF to be able to perform different management functions which are listed in the SFR.
- FMT_MSA.2 requires that the TSF only accepts secure values for the security attributes that are listed in the SFR.
- FMT_MSA.4/AUTH defines rules to configure the object access control with authValue.
- FMT_MSA.1/States requires that a TSF shall enforce the TPM state control SFP to restrict the ability to modify the TOE state.
- FMT_MSA.3/States requires that the TSF shall enforce the TPM state control SFP to provide restrictive default values for security attributes and nobody is authorised to specify alternative default initial values.
- FMT_MSA.1/AC requires that a TSF shall enforce a SFP to restrict the ability to perform dedicated operations on security attributes to dedicated authorised roles..
- FMT_MSA.3/AC requires that the TSF shall enforce the Access Control SFP to provide restrictive default values for security attributes and only dedicated roles are authorised to specify alternative default initial values.
- FMT_MSA.3/NVM requires that the TSF shall enforce the NVM SFP to provide restrictive default values for NVM related security attributes and nobody is authorised to specify alternative default initial values.
- FMT_MSA.4/NVM requires that the TSF shall enforce rules for setting the security attributes of NVM.
- FMT_MSA.1/ExIm requires that a TSF shall enforce the Data Export and Import_SFP to restrict the ability to use the security attribute *authorisation data* for export and import related functions.
- FMT_MSA.3/ExIm requires that the TSF shall enforce the Data Export and Import SFP to provide restrictive default values for security attributes for export and import

related functions. Nobody is authorised to specify alternative default initial values for those security attributes.

- FMT_MSA.1/M&R requires that a TSF shall enforce the Measurement and Reporting SFP to restrict the ability to modify the PCR related security attributes.
- FMT_MSA.3/M&R requires that the TSF shall enforce the Measurement and Reporting_SFP to provide restrictive default values for security attributes for measurement and reporting related functions. Nobody is authorised to specify alternative default initial values for those security attributes.
- FMT_MSA.1/Hier requires that a TSF shall enforce the TPM Object Hierarchy SFP to restrict the ability to modify the security attributes fixedTPM and fixedParent.
- FMT_MSA.3/Hier requires that the TSF shall enforce the TPM Object Hierarchy SFP to provide restrictive default values for security attributes for TPM hierarchy related operations. The creator of the TPM object is authorised to specify alternative default initial values for those security attributes.
- FMT_MSA.4/Hier requires that the TSF shall enforce rules for setting the security attributes of TPM object hierarchies.

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

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

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

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

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

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

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

The security objective **O.Tamper_Leak_Resistance** requires the TOE to resist physical tampering of the TSF by hostile users. This security objective is directly addressed by the SFR FPT_PHP.3 which requires that the TSF resists physical manipulation and physical probing. The objective also requires that the TOE is designed and built in such a way as to control the emanation of intelligible leakage within a specified limit. This security objective is directly addressed by the SFRs FDP_ITT.1 and FPT_ITT.1 that require that the TSF does not emit leakage in excess that could be used to gain access to assets.

7.3.2 Dependency Rationale

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

Table 12: SFR Dependency rationale

SFR	Dependency	Rationale/ fulfilled by
FMT_SMR.1	FIA_UID.1 Timing of identification	Fulfilled by FIA_UID.1
FMT_SMF.1	No dependencies	n. a.
FMT_MSA.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FDP_ACC.1/AC, FMT_MSA.1/AC, FMT_SMR.1
FMT_MSA.4/AUTH	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow	Fulfilled by FDP_ACC.1/Hier,

SFR	Dependency	Rationale/ fulfilled by
	control]	
FCS_RNG.1	No dependencies	n. a.
FIA_SOS.2	No dependencies	n. a.
FMT_MTD.1/AUTH	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled by FMT_SMR.1, FMT_SMF.1
FIA_AFL.1/Recover	FIA_UAU.1 Timing of authentication	Fulfilled by FIA_UAU.1
FIA_UID.1	No dependencies	n. a.
FIA_UAU.1	FIA_UID.1 Timing of identification	Fulfilled by FIA_UID.1
FIA_UAU.5	No dependencies	n. a.
FIA_UAU.6	No dependencies	n. a.
FIA_USB.1	FIA_ATD.1 User attribute definition	Because the TOE does not identify or manage individual users, the SFR FIA_ATD.1 is not applicable here.
FDP_ACC.2/States	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/States
FDP_ACF.1/States	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.2/States, FMT_MSA.3/States
FMT_MSA.1/States	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled by FDP_ACC.2/States, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/States	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/States, FMT_SMR.1
FPT_TST.1	No dependencies	n. a.
FDP_ITT.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/AC, FDP_ACC.1/NVM, FDP_ACC.1/ExIm
FPT_ITT.1	No dependencies	n. a.
FDP_ACC.1/AC	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/AC
FDP_ACF.1/AC	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/AC, FMT_MSA.3/AC
FMT_MSA.1/AC	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles	Fulfilled by FDP_ACC.1/AC, FMT_SMR.1, FMT_SMF.1

SFR	Dependency	Rationale/ fulfilled by
	FMT_SMF.1 Specification of Management Functions	
FMT_MSA.3/AC	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/AC, FMT_SMR.1
FDP_UCT.1/AC	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FTP_ITC.1/AC, FDP_ACC.1/AC
FTP_ITC.1/AC	No dependencies	n. a.
FCS_CKM.1**/**PK	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/AES, FCS_COP.1/ASYMMED, FCS_COP.1/Sign FCS_CKM.4
FCS_CKM.1/ASYMM	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/ASYMMED, FCS_COP.1/Sign FCS_CKM.4
FCS_CKM.1/SYMM	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/AES FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	Fulfilled by FCS_CKM.1**/**PK FCS_CKM.1/ASYMM FCS_CKM.1/SYMM
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	The dependencies regarding key generation/destruction are not applicable here because hash functions do not use any keys.
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 by FCS_CKM.1/PK, FCS_CKM.1/SYMM, FCS_CKM.4
FCS_COP.1/ASYMMED	[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/ASYMM, FCS_CKM.1**/**PK FCS_CKM.4

SFR	Dependency	Rationale/ fulfilled by
FCS_COP.1/Sign	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_CKM.1/PK, FCS_CKM.1/ASYMM FCS_CKM.4
FCS_COP.1/AES	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_CKM.1/PK, FCS_CKM.1/SYMM, FCS_CKM.4
FDP_ACC.1/NVM	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/NVM
FDP_ACF.1/NVM	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/NVM, FMT_MSA.3/NVM
FMT_MSA.3/NVM	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_SMR.1
FMT_MSA.4/NVM	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/NVM
FDP_ITC.1/NVM	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/NVM, FMT_MSA.3/NVM
FDP_ETC.1/NVM	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/NVM
FDP_ACC.1/ExIm	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/ExIm
FDP_ACF.1/ExIm	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/ExIm, FMT_MSA.3/ExIm
FMT_MSA.1/ExIm	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled by FDP_ACC.1/ExIm, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/ExIm	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/ExIm, FMT_SMR.1
FDP_ETC.2/ExIm	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/ExIm

SFR	Dependency	Rationale/ fulfilled by
FDP_ITC.2/ExIm	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	Fulfilled by FDP_ACC.1/ExIm, see rationale (1) and (2) below this table
FDP_UCT.1/ExIm	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/ExIm, see rationale (2) below this table
FDP_UIT.1/ExIm	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	Fulfilled by FDP_ACC.1/ExIm, see rationale (2) below this table
FDP_ACC.1/M&R	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/M&R
FDP_ACF.1/M&R	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/M&R, FMT_MSA.3/M&R
FMT_MSA.1/M&R	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled by FDP_ACF.1/M&R, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3/M&R	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/M&R, FMT_SMR.1
FCO_NRO.1/M&R	FIA_UID.1 Timing of identification	Fulfilled by FIA_UID.1
FDP_RIP.1	No dependencies	n. a.
FPT_FLS.1/FS	No dependencies	n. a.
FPT_FLS.1/SD	No dependencies	n. a.
FPT_PHP.3	No dependencies	n. a.
FDP_SDI.1	No dependencies	n. a.
FDP_ACC.1/Hier	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1/Hier
FDP_ACF.1/Hier	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1/Hier, FMT_MSA.3/Hier
FMT_MSA.1/Hier	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles	Fulfilled by FDP_ACC.1/Hier, FMT_SMR.1, FMT_SMF.1

SFR	Dependency	Rationale/ fulfilled by
	FMT_SMF.1 Specification of Management Functions	
FMT_MSA.3/Hier	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1/Hier, FMT_SMR.1
FMT_MSA.4/Hier	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Fulfilled by FDP_ACC.1/Hier,

The rationales for dependencies that are not fulfilled are:

(1) The SFR FDP_ITC.2/ExIm addresses export and import of user data with security attributes. Data consistency is ensured because the TOE's peer in the data exchange is always a system that is equivalent to the TOE. In particular, the same TPM may import exported data. Hence the SFR FPT_TDC.1 is not applicable.

(2) The exported and imported user data is based on data objects and not channel-based. The security attributes are part of the exported object and the object is integrity and confidentiality protected. Hence the SFRs regarding trusted channel and trusted path are not applicable.

7.3.3 Assurance Rationale

This protection profile requires the TOE to be evaluated on Evaluation Assurance Level 4 (EAL4) as defined in CC [3] and augmented with ALC_FLR.1 and AVA_VAN.4 listed in table 10.

EAL4 was selected because the objective of the TOE is to provide developers or users with a moderate to high level of independently assured security in conventional commodity TOEs and assumes that developers or users are prepared to incur additional security-specific engineering costs. EAL4 permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources.

The developer and manufacturer ensure that the TOE is designed and fabricated so that the TSF achieves the desired properties and it requires a combination of equipment, knowledge, skill, and time to be able to derive design information or affect the development and manufacturing process in a way that could compromise security through attack. This is addressed by the SAR of the class ALC especially by the component ALC_DVS.1.

Further the AVA_VAN.4 requires the developer and the manufacturer to provide necessary evaluation evidence that the TOE fulfills its security objectives and is resistant to attack with **Moderate** potential. The component AVA_VAN.4 will analyze and assess the resistance of the TOE to attacks with **Moderate** attack potential.

EAL4 is also augmented with ALC_FLR.1 to track and correct both reported and found security flaws in the product.

The component AVA_VAN.4 Methodical vulnerability analysis has the following dependencies:

ADV_ARC.1 Security architecture description
ADV_FSP.2 Security-enforcing functional specification

ADV_TDS.3 Basic modular design
ADV_IMP.1 Implementation representation of the TSF
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

All these components are contained in the EAL4 package. The component ALC_FLR.1 Basic flow remediation has no dependencies. Therefore all these dependencies are satisfied by EAL4.

8. Appendix

8.1 Field Upgrade Module

8.1.1 Introduction

The TCG TPM2.0 Automotive Thin Profile [33] describes optional commands (TPM2_FieldUpgradeStart and TPM2_FieldUpgradeData) which may be supported by Automotive-Thin TPMs capable of upgrading their own software after being deployed in the Field. This PP-Module provides a functional package to support the ST writer when describing security functional requirements for TPM firmware Field Upgrade. This PP-Module supplements the requirements of the Base PP, the Protection Profile Automotive-Thin Specific Trusted Platform Module Specification Family 2.0. Please note that TPM firmware upgrade is the process of uploading new software inside the TPM. This process does not concern the ECU platform firmware upgrade which is a process proper to the ECU.

8.1.2 TPM Life cycle

In the case that Field Upgrade is supported by the Automotive Thin TPM, the TPM life cycle may distinguish a second case in addition to case1 described in chapter 2.2.4 of this PP. The ST writer may add the following content in the chapter 2.2.4:

Because the current Automotive Thin TPM supports optional Field Upgrade, the TPM life cycle distinguishes a second case,

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

The full Field Upgrade (cf. [77], chapter 12.5.2) does and the incremental Field Upgrade may change the TOE. The TPM life cycle is linked also to the life cycles of the EPS, PPS and SPS (if supported) and their corresponding key hierarchies.

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

The following steps describe the life cycle case 2 for the upgraded firmware parts only (see Figure 5). The TOE hardware is as already delivered to the platform vendor or the end user.

- Development of TPM (Phase 1)

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

- Manufacturing of the TPM (Phase 2)

The TPM manufacturer delivers the firmware upgrade for Field Upgrade to the platform vendor as their customer.[165]

- Platform Integration (Phase 3)

  The platform vendor uses the Field Upgrade functionality[166] to install the new TPM firmware on hardware of a previous TPM before delivery of the platform to the end user.

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

- Operational usage (Phase 4)

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

  The Field Upgrade preserves User data (NV index allocations and contents, persistent object allocations and contents) and TSF data (EPS and PPS, hierarchy authentication reference data *authValue* and *proof*, *lockoutCount*, the PCR *authValue* values). After Field Upgrade the new TPM will be ready for operational use in the environment of the end user.

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

[165] The TPM manufacturer may use the field upgrade process as well but this is not expected due to the intended supply chain of TPM.

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

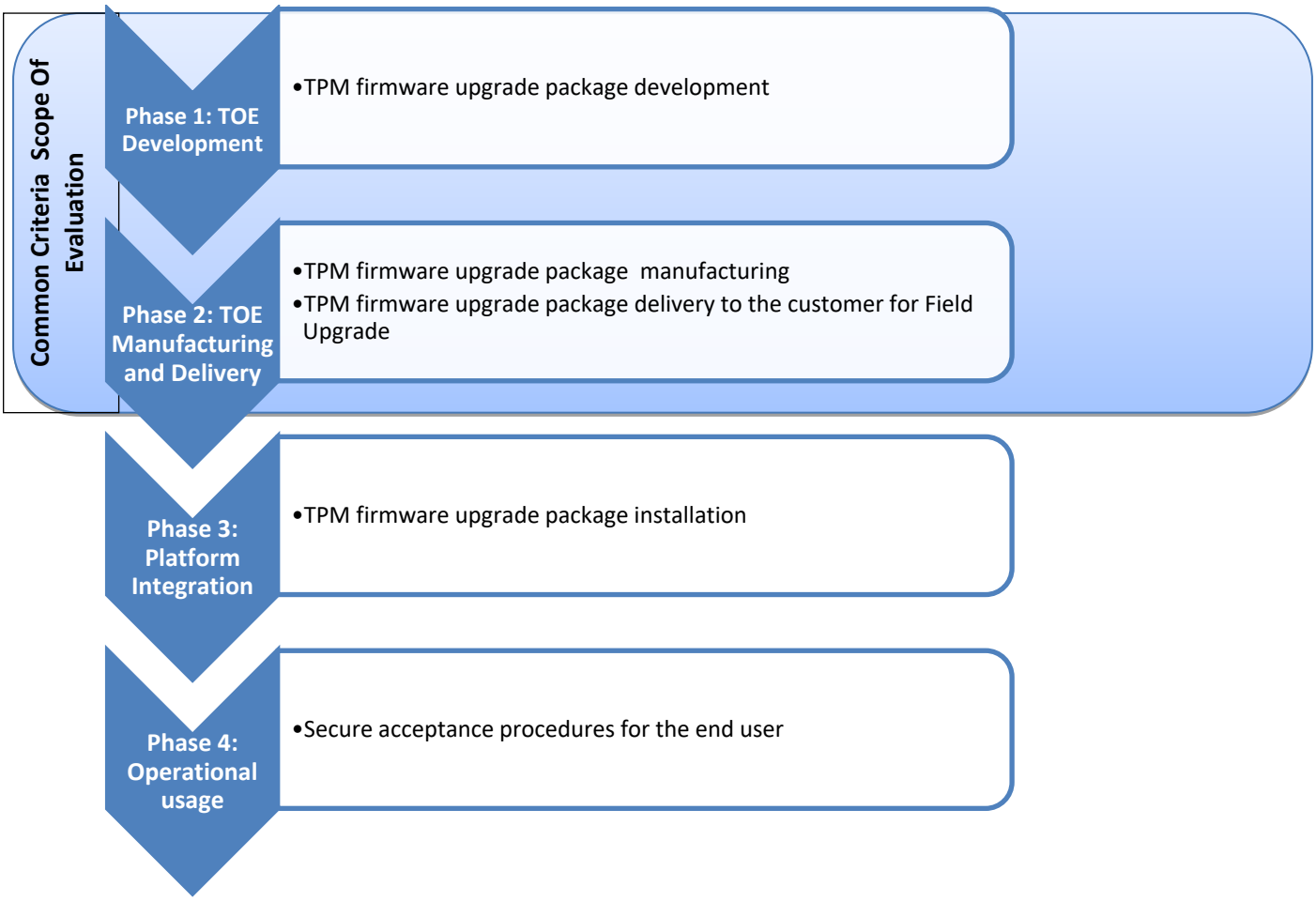


Figure 5 TPM Life Cycle case 2

8.1.3 Conformance Claims

The following sections describe the conformance claims of the Field Upgrade PP-Module.

8.1.3.1 CC Conformance Claim

This PP-Module claims to be conformant with the Common Criteria version 3.1 Release 5 as follows

- Part 2 extended,
- Part 3 conformant.

8.1.3.2 Conformance with Packages

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

8.1.3.3 Conformance with other Protection Profiles

Conformance to this PP-Module requires conformance to the requirements in the Protection Profile Automotive-Thin Specific Trusted Platform Module Specification Family 2.0.

8.1.3.4 Conformance Statement

This PP-Module requires **strict** conformance of any Security Target (ST) or PP that claims conformance to the base PP and this PP-Module.

8.1.4 Threats

There are no threats specific to this PP-Module. The ST writer may relate PP-module functionality to the existing threats in Table 1.

8.1.5 Organisational Security Policies

The ST writer may introduce a new OSP in Table 2 as follows:

4	OSP.FieldUpgrade	The Platform software is allowed to perform Field Upgrade within the certified TPM before and after delivery to the end user. The end user shall be made aware of the certification and the version of the TPM.

8.1.6 Security Objectives

The ST writer may introduce a new Objective for the TOE in Table 4 as follows:

19	O.FieldUpgradeControl	The TOE restricts the Field Upgrade to authorised role and accepts only authentic update data provided by the TOE vendor.

The ST writer may introduce a new Objective for the Operational Environment in Table 5 as follows:

4	OE.FieldUpgradeInfo	The developer via AGD documentation will instruct the admininstrator doing the upgrade how to do the upgrade and that the admininstrator should inform the end user regarding the Field Upgrade process, its result, and the version of the certified TPM.

The ST writer may complete Table 6 and may introduce a new Security Objective rational in chapter 5.3 from Table 13 as follows:

Table 13 FU Security Objective Rationale

	O.FieldUpgradeControl	OE.FieldUpgradeInfo
OSP.FieldUpgrade	X	X

OSP.FieldUpgrade**:** The Platform software is allowed to perform Field Upgrade within the certified TPM before and after delivery to the end user. The end user shall be made aware of the certification and the version of the TPM.

The **OSP.FieldUpgrade** is implemented by O.FieldUpgradeControl and OE.FieldUpgradeInfo:

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

8.1.7 Assumptions

The ST writer may update Table 3 as follows:

#	Assumption	Description
2	A.ConfigurationFU	The Field Upgrade package will be properly installed and configured based on AGD instructions.

8.1.8 Security Requirements

The ST writer may update the chapter 7.1.4.1 "TPM Operational states" as follows in order to explain the new state diagram of the TOE, the new Objects/Operations/Attributes, and the new security functional requirements needed for support of Field Upgrade:

- Initialisation state: The TPM enters this state when it receives the _TPM_Init indication. This indication is provided in a platform-specific manner (cf. section 12.2.2 of [7] for details). In the Init state, only the commands TPM2_Startup and field upgrade are accepted. All other commands do not change the state and imply an error return code. The TPM may perform self-test in the Init state and may enter Failure mode if the self-test detects any failure.
- FUM: The Field Upgrade Mode is described in the specification [7] in section 12.5 as an optional and vendor specific capability for upgrading the TPM firmware. The specification does not define the detailed behavior of Field Upgrade Mode and allows vendor specific implementation. According to the library specification the TPM enters the FUM from operational or Init state after receiving the command TPM2_FieldUpgradeStart and successful integrity and authenticity validation of the first upgrade data block, accepts TPM2_FieldUpgradeData commands only in FUM and exits FUM returning to normal operation or entering a mode that requires _TPM_Init before normal operations resume. The Field Upgrade Mode can also be reached after TPM_Init if Field Upgrade loading process has been interrupted and needs to be resumed before the TPM returns to operational state. The TPM shall perform integrity and authenticity checks, but may implement vendor specific authorisation or vendor specific commands and related state transitions for FUM. The informative Figure 3 denotes these possible state transitions with dashed lines.

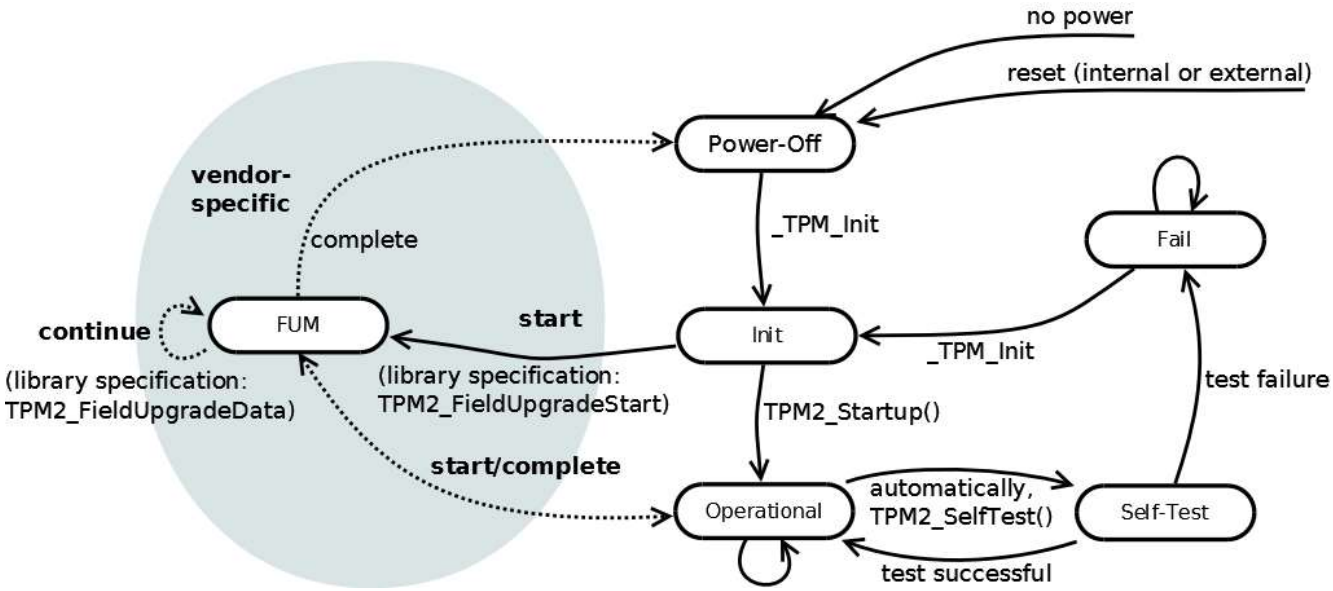


Figure 6: States of the TPM and its Transitions (informative)

Application note 33: Figure 6 illustrates the transitions between the TPM operational states as defined in the library specification, chapter 12 of [7]. The Field Upgrade Mode is vendor specific. The state transition and the commands TPM2_FieldUpgradeStart and TPM2_FieldUpgradeData shown in Figure 3 as described in the library specification are optional.

The ST writer may update Table 9 to add the following objects, operations, and security attributes:

Table 14: objects, operations and security attributes for the TPM state control SFP

#	Protected Objects	Operations	Security attributes
2	**Firmware update data** Data provided by the vendor in order to replace the firmware or parts of the firmware.	**TPM2_FieldUpgradeStart()** Entering FUM and accepting the first data block of Firmware update data **TPM2_FieldUpgradeData()** Read the following Firmware update data blocks.	Authorisation data for TPM2_FieldUpgradeStart(): **platformAuth:** hierarchy authorisation to change auth. Security attributes of firmware update data: **Signature** over the digest of the first or the complete Firmware update data, generated by the TPM manufacturer **Digest** over each block or the complete Firmware update data

The ST writer may describe the security functional requirements for FieldUpgrade as follows:

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

Hierarchical to: FDP_ACC.1 Subset access control
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1/FU The TSF shall enforce the TPM Field Upgrade SFP[167] on all subjects and objects[168] and all operations among subjects and objects covered by the SFP.

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

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

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

FDP_ACF.1.1/FU The TSF shall enforce the TPM Field Upgrade SFP[169] to objects based on the following
Subjects as defined in Table 7:
(1) Platform firmware with the security attributes platformAuth and physical presence if supported by the TOE,
(2) all other subjects; their security attributes are irrelevant for this SFP,
Objects as defined in Table 8 and Table 9:
(3) Firmware update data with security attributes signature of the TPM manufacturer and digest,
(4) all other objects; their security attributes are irrelevant for this SFP[170].

FDP_ACF.1.2/FU The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
(1) The [assignment: *authorised role*] is authorised to change the TPM state to FUM if the authenticity of the first digest or the signature could be successfully verified.
(2) While in FUM state the Platform firmware is authorised to import or activate firmware data only after successful verification of its integrity and authenticity (see FDP_UIT.1/FU).
(3) The FUM state shall only be left when [assignment: *rules for a state transition from FUM to another state*].

---

[167] [assignment: *access control SFP*]
[168] [assignment: *list of subjects and objects*]
[169] [assignment: *access control SFP*]
[170] [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.1.3/FU 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/FU The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

Application note 34: The ST writer shall define additional rules in FDP_ACF.1.2/FU for the state transitions while the TPM is in FUM. Section 12.5 of [7] describes optional protected capabilities for upgrading the TPM firmware.

Application note 35: When parts of the TSF or the complete TSF are replaced by a firmware update then the entire TOE needs to be considered as replaced by installation of another TOE.

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

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

FMT_MSA.1.1/FU TSF shall enforce the TPM Field Upgrade SFP[171] to restrict the ability to modify[172] the security attributes TPM state
(1) FUM[173] to Platform firmware[174],

Application note 36: The concrete restrictions in the TPM Field Upgrade SFP to restrict the modification of the TPM state by dedicated roles is defined in FMT_MSA.1/FU.

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

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

FMT_MSA.3.1/FU The TSF shall enforce the TPM Field Upgrade SFP[175] to provide restrictive[176] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/FU The TSF shall allow ~~the~~ nobody[177] to specify alternative initial values to override the default values when an object or information is created.

---

171 [assignment: *access control SFP(s), information flow control SFP(s)*]
172 [selection: *change_default, query, modify, delete, [assignment: other operations]*]
173 [assignment: *list of security attributes*]
174 [assignment: *the authorised identified roles*]
175 [assignment: *access control SFP, information flow control SFP*]
176 [selection, choose one of*: restrictive, permissive, [assignment: other property]*]

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

Hierarchical to: No other components.

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

FDP_UIT.1.1/FU The TSF shall enforce the TPM Field Upgrade SFP[178] to receive[179] ~~user~~ **firmware update** data in a manner protected from [selection: *modification, deletion, insertion, replay*] errors.

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

8.1.9 Security Requirements rationale

The ST writer may update Table11 with the following:

Table 15 FU Security Requirements Rationale

	O.Import	O.FieldUpgradeControl
FMT_SMR.1		x
FIA_UAU.5		x
FDP_ACC.2/FU		x
FDP_ACF.1/FU		x
FMT_MSA.1/FU		x
FMT_MSA.3/FU		x
FDP_UIT.1/FU	x	x

[177] [assignment: *the authorised identified roles*]
[178] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[179] [selection: *transmit, receive*]

The ST writer may introduce a new Sufficiency of SFR rational for O.FieldUpgradeControl in chapter 7.3.1 as follows:

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

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

The ST writer may complete the Sufficiency of SFR rational for **O.Import** in chapter 7.3.1 as follow:

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

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

The ST writer may update Table 12 in chapter 7.3.2 as follow:

Table 16: SFR Dependency rationale

SFR	Dependency	Rationale/ fulfilled by
FDP_ACC.2 / FU	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1 / FU
FDP_ACF.1 / FU	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.2 / FU,

SFR	Dependency	Rationale/ fulfilled by
		FMT_MSA.3 / FU
FMT_MSA.1 / FU	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled by FDP_ACC.2 / FU, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3 / FU	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1 / FU, FMT_SMR.1
FDP_UIT.1 / FU	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	Fulfilled by FDP_ACC.2 / FU, see rationale (1) below this table

(1) The firmware update procedure as a kind of user data import is achieved via commands that transfer single data packets into the TOE. No secure channel will be established and used for that process: the protection of the user data is done based on checks of each single packet. Hence the SFRs regarding trusted channel and trusted path are not applicable.

8.2 Random Number Generator (informative)

The internal RNG shall comply with the NIST Special Publication 800-90A [17]. Hence, its primary nature is that of a deterministic random bit generator (DRBG). According to [17] the entropy is taken from a seed given as input to the DRBG mechanism. The DRBG can be reseeded in order to add new entropy to the internal state.

In the TPM architecture specification [7], the RNG architecture is given in section 11.4.10. As shown in figure 4 of [7], the RNG contains (at least) one entropy source in order to seed or reseed the DRBG. The entropy should be collected in a state register of the RNG that is not visible to an outside process or other TPM capability. Using the command TPM2_StirRandom, additional data can be injected into the status registers, but the security of the DRBG itself does not rely on the secrecy of this information.

In order to meet the certification requirements of the intended market, the quality metric of the RNG depends in part on the quality of the entropy source and the seed period: If the seeding takes place only on initialisation time, the resulting RNG is a pure deterministic RNG. On the other hand, if the reseeding mechanism ensures that the entropy inserted into the RNG always exceeds the amount of entropy that is taken as output from the RNG, the RNG can be seen as physical RNG. Also, the reseeding could be implemented on a periodic base. In that case the amount of output data taken from the RNG may be bigger than the amount of entropy that was injected by reseeding. In that case the character of the RNG is hybrid. In summary, the character of the RNG can be determined by choosing the seed period: An infinite seed period creates a deterministic RNG while a very short seed period creates a physical RNG.
NIST Special Publication 800-90B [18] includes guidance about the quality of entropy sources and determination of the amount of entropy provided by an entropy source.

8.3 Acronyms

For the purposes of this document, the acronyms given in CC Parts 2 and 3 and the following apply.

Table 17: Acronyms

Acronym	Description
TPM	Prefix for an indication passed from the system interface of the TPM to a Protected Capability defined in the TPM 2.0 Library specification
AuthData	Authentication Data or Authorisation Data, depending on the context
CA	Certificate Authority
CFB	Cipher Feedback mode
CRTM	Code Root of Trust for Measurement
CTR	Counter-mode encryption
DA	Dictionary Attack
DAA	Direct Autonomous Attestation
DRBG	Deterministic Random Bit Generator
EAL	evaluated assurance level
ECB	Electronic Codebook
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECU	Embedded Control Unit
EK	Endorsement Key
EPS	Endorsement Primary Seed
FIPS	Federal Information Processing Standard
FUM	Field Upgrade mode
HMAC	Hash Message Authentication Code
HW	Hardware Interface
I/O	Input/Output
IV	Initialisation Vector
KDF	key derivation function
MMIO	Memory Mapped I/O
NIST	National Institute of Standards and Technology
NV	Non-volatile
NVM	Non-Volatile Memory
OAEP	Optimal Asymmetric Encryption Padding
OSP	Organisational Security Policies
PCR	platform configuration register(s)
PK	Primary Key
PP	Physical Presence, Protection Profile
PPO	Platform Primary Object
PPS	Platform Primary Seed
PRIVEK	Private Endorsement Key

Acronym	Description
PRNG	Pseudo Random Number Generator
PUBEK	Public Endorsement Key
RNG	Random Number Generator
RSA	Algorithm for public-key cryptography. The letters R, S, and A represent the initials of the first public describers of the algorithm Rivest, Shamir and Adleman.
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SPS	Storage Primary Seed
SRK	Storage Root Key
ST	Security Target
TCB	trusted computing base
TCG	Trusted Computing Group
TOE	Target of Evaluation
TPM	Trusted Platform Module
TPM2_	Prefix for a command defined in the TPM 2.0 Library specification
TSF	TOE Security Functions
UTC	Universal Time Clock

8.4 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 5, CCMB-2017-04-001, April 2017

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 3.1, Revision 5, CCMB-2017-04-002, April 2017

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 5, CCMB-2017-04-003, April 2017

[4] Common Methodology for Information Technology Security Evaluation Methodology, Evaluation Methodology, Version 3.1, Revision 5, CCMB-2017-04-004, April 2017

[5] Common Criteria Recognition Arrangement Management Committee, Policies and Procedures, Supporting Documents for Smartcards and similar devices, document number 2006-06-001

[6] Supporting Document Guidance Smartcard Evaluation, February 2010, Version 2.0, CCDB-2010-03-001

[7] TPM Library Part 1: Architecture, Specification Version 2.0, Revision 1.38, September 2016, Trusted Computing Group, Incorporated

[8] TPM Library Part 2: TPM Structures, Specification Version 2.0, Revision 1.38, September 2016, Trusted Computing Group, Incorporated

[9] TPM Library Part 3: Commands, Specification Version 2.0, Revision 1.38, September 2016, Trusted Computing Group, Incorporated

[10] TPM Library Part 4: Supporting Routines, Specification Version 2.0, Revision 1.38, September 2016, Trusted Computing Group, Incorporated

[11] ITU-T X.690: Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

[12] FIPS-140-2, Federal Information Processing Standard 140-2

[13] FIPS-180-4, Federal Information Processing Standard 180-4 Secure Hash Standard (SHS)

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

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

[16] FIPS-197, Federal Information Processing Standard 197

[17] NIST Special Publication 800-90A: Recommendation for Random Number Generation Using Deterministic Random Bit Generators. January 2012

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

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

[20] NIST Special Publication 800-107: Recommendation for Applications Using Approved Hash Algorithms. August 2012

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

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

[23] IETF RFC 2104, Internet Engineering Task Force Request for Comments 2104: HMAC: Keyed-Hashing for Message Authentication

[24] IETF RFC 2119, Internet Engineering Task Force Request for Comments 2119: Key words for use in RFCs to Indicate Requirement Levels

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

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

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

[28] ISO/IEC 10118-3, Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash function

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

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

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

[32] TCG TPM 2.0 Automotive Thin Profile Version 1.01 revision 14, December 08 2017, Trusted Computing Group, Incorporated