SLB96xx

Security Target

SOLID FLASH™

Version: 1.3

Date: 2017-06-28

Autor: Jürgen Noller

PUBLIC

Chip Card & Security

Edition 1.3

Published by

Infineon Technologies AG

81726 Munich, Germany

Revision History	
Page or Item	**Subjects (major changes since previous revision)**
1.3	**2017-06-28 New firmware versions added**
1.2	2016-12-23 Firmwareversion v4.42.0132.00 and v4.42.0133.00 added
1.1	2013-12-02 Update of 7.1.1
1.0	2013-10-25 Final Version

TABLE OF CONTENTS

1 Security Target Introduction

This section contains the document management and provides an information overview. The Security Target (ST) identification provides the labelling and descriptive information necessary to identify, catalogue, register, and cross-reference a ST. The ST overview summarizes the profile in narrative form and provides sufficient information for a potential user to determine whether the ST is of interest. The overview can also be used as a standalone abstract for ST catalogues and registers.

1.1 Security Target and Target of Evaluation Reference

The title of the security target (ST) is SLB96xx Security Target.
The security target has the version 1.3 and is dated 2017-06-28.

The Target of Evaluation (TOE) is a security IC (Security Controller) with integrated firmware (operating system) and guidance documentation, which is named SLB96xx, is internally registered under the development code v4.43.0257.00 and v4.43.0258.00 and v4.43.0259.00.

The Security Target is based on the Trusted Computing Group Protection Profile PC Client Specific Trusted Platform Module Family 1.2; Level 2 Revision 116 Version: 1.2 (PP, [8]).
The Protection Profile and the Security Target are built in accordance with Common Criteria V3.1.

	Version	Date	Registration
Security Target	1.3	2017-06-28	SLB96xx Security Target
Target of Evaluation			SLB96xx
Firmware	v4.43.0257.00, v4.43.0258.00, v4.43.0259.00		v4.43.0257.00 and v4.43.0258.00 and v4.43.0259.00
Protection Profile	1.2	2011-05-18	Trusted Computing Group Protection Profile PC Client Specific Trusted Platform Module Family 1.2; Level 2, Revision 116 BSI-CC-PP-0030-2008-MA-01
Guidance Documentation	V1.2 Rev. 116	2011-03-01	TPM Main Part 1 Design Principles, Specification Version 1.2 Revision 116, Trusted Computing Group
	V1.2 Rev. 116	2011-03-01	TPM Main Part 2 TPM Structures Specification Version 1.2 Revision 116 Trusted Computing Group
	V1.2 Rev. 116	2011-03-01	TPM Main Part 3 Commands Specification Version 1.2 Revision 116 Trusted Computing Group
	V1.21 Rev. 1.00	May 2011	TCG PC Client Specific TPM Interface Specification (TIS) for TPM Family 1.2; Level 2, Version 1.21 Final, Revision 1.00
	Revision 1.4	2017-06-26	OPTIGA™ TPM SLB 9660 TPM1.2 Databook
	Rev. 1.00	Edition 2013-03	TPM Trusted Platform Module Version 1.2 SLB9660 Basic Platform Manufacturer Guideline
	Rev. 1.9	2017-06-26	OPTIGA™ TPM SLB 9660 TPM1.2 Errata and Updates
Common Criteria	3.1 Rev 3	2009-July	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2012-09-001 Part 2: Security functional requirements CCMB-2012-09-002 Part 3: Security Assurance Components CCMB-2012-09-003

Table 1: Identification

Remarks to the Target of Evaluation (TOE):

The TOE of this Security Target encloses following different versions: v4.43.0257.00 and v4.43.0258.00 and v4.43.0259.00. All of these versions may include different derivatives. The hardware and software of these derivatives are identical, the only difference between the derivatives are the extended temperature range, the packaging and the own intermediated IFX certificate. The versions v4.43.0257.00 and v4.43.0258.00 and v4.43.0259.00 including the identical source code, which the v4.43.0258.00 and v4.43.0259.00 are used for field upgrade. The derivatives are listed in the document "OPTIGA™ TPM SLB 9660 TPM1.2 Errata and Updates" [88] in section 5 " Sales Order Code".

1.2 Target of Evaluation Overview

This Security Target (ST) describes the target of evaluation (TOE) known as the Infineon SLB96xx Trusted Platform Module (TPM) and gives a summary product definition. In the following description the expressions SLB96xx or TPM stands for all the TOE derivatives of the TOE.

The SLB96xx Trusted Platform Module, called TPM or SLB96xx in the following text, is an integrated circuit and software platform that provides computer manufacturers with the core components of a subsystem used to assure authenticity, integrity and confidentiality in e-commerce and internet communications within a Trusted Computing Platform. The SLB96xx is a complete solution implementing the version 1.2 of the TCG Trusted Platform Module Main, Specification Version 1.2, [5] [6] [7] and the TCG PC Client Specific TPM Interface Specification, Version 1.2 Final, Revision 1.00 [9].

The SLB96xx uses the LPC interface (Low Pin Count) as defined by Intel for the integration into existing PC mainboard. The SLB96xx is basically a secure controller with the following added functionality:

- Random number generator (DRNG)
- Asymmetric key generation (RSA keys with key length up to 2048 bit)
- Symmetric key generation (AES keys, for internal use only)
- Symmetric and asymmetric key procedures (encryption/decryption, generation and verification of digital signatures)
- Hash algorithms (SHA-1) and MAC (HMAC)
- Secure key and data storage
- Identification and Authentication mechanisms
- Tick and Monotonic Counter

In this security target the TOE (target of evaluation) is described and a summary specification is given. The security environment of the TOE is defined. The assets are identified which have to be protected through the security policy. The threats against these assets are described. The security objectives as the objectives of the security policy are defined as well as the security requirements. The applicable IT security requirements are taken from the Common Criteria, with appropriate refinements. The security requirements are constructed out of the security functional requirements as part of the security policy and the security assurance requirements, as the steps during the evaluation and certification to prove that the TOE meets these requirements. The functionality of the TOE to meet the requirements is described.

The assets, threats, security objectives and the security functional requirements are defined in the Trusted Computing Group Protection Profile PC Client Specific Trusted Platform Module TPM Family 1.2; Level 2 Revision 116; Version: 1.2 [8], and are referenced here.

The TOE summary specification consisting of the security features, the assurance requirements and the security function policies are defined in the ST as property of this specific TOE, the SLB96xx. The rationale presents evidence that the ST is a complete and cohesive set of requirements and that a conformant TOE would provide an effective set of IT security countermeasures within the security environment.

2 Target of Evaluation Description

The TOE description helps to understand the specific security environment and the security policy. In this context the assets, threats, security objectives and security functional requirements can be employed. The following is a more detailed description of the TOE than in the Trusted Computing Group Protection Profile [8] as it belongs to the specific TOE.

2.1.1 TOE Definition

The Target of Evaluation (TOE) is the “Infineon SLB96xx Trusted Platform Module” of the Infineon Technologies AG called “SLB96xx” or “TPM” in the following description. The TOE is an integrated circuit and software platform that provides computer manufacturers with the core components of a subsystem used to assure authenticity, integrity and confidentiality in e-commerce and internet communications within a Trusted Computing Platform as defined in the TPM Main Specification. The SLB96xx is a complete solution implementing the TCG TPM Main Specification Version 1.2, [5] [6] [7] and the TCG PC Client Specific TPM Interface Specification (TIS) for TPM Family 1.2; Level 2, Version 1.21 Final, Revision 1.00 [9].

A Trusted Platform is a platform that can be trusted by local users and by remote entities. The basis for trusting a platform is a declaration by a known authority that a platform with a given identity can be trusted to measure and report the way it is operating. That operating information can be associated with data stored on the platform, to prevent the release of that data if the platform is not operating as expected. Other authorities provide declarations that describe the operating information the platform ought to produce when it is operating properly. The local user and remote entities trust the judgment of the authorities; so, when they receive proof of the identity of the platform, information about the current platform environment, and proof about the expected platform environment, they can decide whether to trust the platform to behave in a sufficiently trustworthy and predictable manner. The local user and/or remote entities must take this decision themselves because the level of trust in a platform can vary with the intended use of that platform, and only the local user and/or remote entities know that intended purpose.

The trusted mechanism of the platform uses cryptographic processes, including secrets. The trusted mechanisms are required to be isolated from the platform in order to protect secrets from disclosure and protect methods from subversion.

The subsystem protects itself against physical and software attacks to provide protection against attacks to the platform.

Some, but not all, subsystem capabilities must be trustworthy for the subsystem to be trustworthy. These are called the “Trusted Set” (TS). Other capabilities must work properly if the subsystem is to work properly, but they do not affect the level of trust in a Subsystem. These are called the “Trusted platform Support Set” (TSS).

The Trusted Set of capabilities can be partitioned into measurement capabilities, reporting capabilities, and storage capabilities. The trusted measurement capabilities are called the “Root of Trust for Measurement” (RTM). The trusted reporting capabilities are called the “Root of Trust for Reporting” (RTR). The trusted storage capabilities are called the “Root of Trust for Storage” (RTS). The RTM makes reliable measurements about the platform and puts the measurement results into the RTR. The RTR prevents unauthorized changes to the measurement results, and reliably reports those measurement results. The RTS provides methods to minimize the amount of trusted storage that is required. The “Root of Trust for Measurement” and the “Root of Trust for Reporting” cooperate to permit an entity to believe measurements that describe the current computing environment in the platform. An entity can assess those measurement results and compare them with values that are to be expected if the platform is operating as expected. If there is sufficient match between the measurement results and the expected values, the entity can trust computations within the platform (not just within the TS) to execute as expected.

The RTR have a cryptographic identity in order to prove to a remote entity that RTR messages come from genuine trusted capabilities, and not from bogus trusted capabilities.

The SLB96xx is basically a secure controller with the following added TOE security services (TOSS):

Random number generation (DRNG)

The DRNG capability is only accessible for valid TPM commands. Intermediate results from the DRNG are not available to any user. When the data is for internal use by the TPM (e.g. asymmetric key generation) the data is held in a shielded location and is not accessible to any user.

Algorithms: RSA, SHA-1, AES, MGF1

The SLB96xx supports the RSA algorithm for encryption and digital signatures with key sizes of 512, 1024 and 2048 bits. The RSA implementation provides protection and detection of failures during the Chinese Remainder Theorem (CRT) process. The TPM storage keys and TPM identity keys are of strength equivalent to a 2048 bit RSA key. A storage key whose strength is less than that of a 2048 RSA key could not be stored in the SLB96xx. The TPM identity keys are RSA keys with key size of 2048 bits.

The RSA algorithm is used for signature and verification operations according PKCS#1 V2 for the format and design of the signature output.

The SLB96xx supports the Secure Hash Algorithm-1 (SHA-1) hash algorithm as defined by United States Federal Information Processing Standard 180-2. The output of the SHA-1 is a 160 bit hash value and all areas that expect a hash value are required to support the full 160 bits.

The SLB96xx supports the AES algorithm with key sizes of 128 bits for encryption and decryption. The function is used to support the transportation functionality and to support the temporary cashing of keys outside the TPM and for the personalisation of code and data.

The SLB96xx supports the MGF1 algorithm with key sizes of 160 bits for encryption and decryption. The function is used to support the transportation functionality.

Key Generation

The SLB96xx generates asymmetric key pairs (algorithm RSA) in accordance with P1363 and AES keys with key sizes of 128 bits (for internal use). The generation function is a protected capability and the private key and the AES key is held in a shielded location.

For the HMAC key generation and for the creation of all nonce values the next n bits are taken from the internal TPM DRNG based on NIST standard..

The keys managed by the TPM may be non-migratable or migratable. A non-migratable key is a key that cannot be transported outside a specific TPM. A migratable key is a key that may be transported outside the specific TPM. The certified migratable key allows for migration but still has properties that the TPM can certify.

Self -Tests

The SLB96xx provides startup self-tests and a mechanism to allow self-tests to be run on demand. The response from self-tests is pass or fail. Self-tests include checks of the following:

- RNG functionality (according [11] class DRG.3).
- Reading and extending the integrity registers.
- Endorsement key pair integrity. This test verifies the RSA sign and verify engine by signing and verifying a known value with the endorsement key pair.
- Integrity of the protected capabilities of the SLB96xx.

- Test of the tamper-resistance markers.

If a failure during any self-test is detected, the part experiencing the failure will enter a shutdown mode and an error code is returned.

Identification and Authentication

The TPM identification and authentication capability is used to authenticate an entity owner and to authorize use of an entity. The basic premise is to prove knowledge of a shared secret. This shared secret is the identification and authentication data. The TCG specification calls the identification and authentication process and this data authorization.

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

The identification and authentication protocols use a random nonce. This requires that a nonce from one side be in use only for a message and its reply. For instance, the TPM would create a nonce and send that on a reply. The requestor would receive that nonce and then include it in the next request. The TPM would validate that the correct nonce was in the request and then create a new nonce for the reply. This mechanism is in place to prevent replay attacks and man-in-the-middle attacks.

Access control

Access control is enforced in the SLB96xx on all data and operations performed on that data. The TPM provides access control by denying access to some data and operations and allowing access to other data and operations based on the value of different flags called security attributes (e.g. TPM_AUTH_DATA_USAGE, TPM_KEY_FLAGS, TPM_KEY_USAGE) which are defined in the PP [8], Table 1. For example the security attribute TPM_AUTH_DATA_USAGE defines access as either “never” or “always”. Always must be authenticated with a shared secret. Never means that usage of the key is permitted by anyone without authentication. The security attribute TPM_KEY_FLAGS defines information regarding a key, e.g. a key is a migratable, non-migratable or certifiable migratable. A non-migratable key is a key that cannot be transported outside a specific TPM. A migratable key is a key that may be transported outside the specific TPM. In addition some keys must be bound to a specific TPM but should be able to be backuped or migrated under certain circumstances. The certified migration allows a Migration Selection Authority therefore to control a migration process without handle the migrated key itself or respectively uses a Migration Authority to control the migration process with the knowledge of the data of the migrated key. Those keys which are planned for certified migration are called certifiable migratable keys. The security attribute TPM_KEY_USAGE identifies the permitted usage of a key. Depending on the key type (e.g. signing key, storage key and identity key) certain operations may or may not be allowed using the particular key. Upon appropriate identification and authentication associated with the keys, users can use the key for the purposes permitted by the security attribute TPM_KEY_USAGE. The security attributes are stored as flags in the TPM or associated with the data in an encrypted blob.

Tick and Monotonic Counter

The SLB96xx provides a “tick counter” as a count of the numbers of ticks that have occurred since the start of a timing session. The time between the ticks is identified by the “tick rate” but it is the responsibility of the caller to associate the ticks to an actual UTC time.

The SLB96xx provides also a monotonic counter as an ever-increasing incremental value for external use.

Root of Trust for Storage

The SLB96xx provides non-volatile storage as shielded location for data of external entities. The TPM owner controls access to the non-volatile storage. The access control may include the need of authentication of the user, delegations, PCR values and other controls. Additionally the SLB96xx has the capability of secure storage for an unlimited number of private keys or other data. The resulting encrypted file, which contains header information in addition to the data or key, is called a blob and is output by the TPM, it can be re-loaded in the TPM when needed. The functionality of the SLB96xx can also be used that private keys generated on the TPM can be stored outside the TPM (encrypted) in a way allowing the TPM to use them later without ever exposing such keys in the clear outside the TPM.

The functionality used to provide secure storage is:

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

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

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

- The Storage Root Key (SRK), which is the root key of a hierarchy of keys associated with a TPM; it is generated within a TPM and is a non-migratable key. Each owned TPM contains a SRK, generated by the TPM at the request of the Owner. Under that SRK may be organized different trees dealing with migatable data or non-migratable data.
- Storage keys, which are used to RSA encrypt and RSA decrypt other keys and their security attributes in the Protected Storage hierarchy, only.
- Binding keys, which are used for TPM_Unbind operations only. A bound operation (performed outside the TPM) associates identification and authentication data with a particular data set and the entire data blob is encrypted outside the TPM using a binding key, which is an RSA key. The TPM_Unbind operation uses a private key stored in the TPM to decrypt the blob so that the data (often a key pair) stored in the blob may be used.

Root of Trust for Reporting

The root of trust for reporting (RTR) exposes the measurement digests stored in the PCRs and attest to the authenticity of these measurement digests based on trusted platform identities or Direct Anonymous Attestation Protocol. The trusted platform identities for RTR are defined by

Attestation Identity Credentials for Attestation Identity Keys (AIK) generated by the TPM. The TPM creates digital signatures over the PCR values using an Attestation Identity Key.

Each SLB96xx is identified and validated by its Endorsement Key. The Endorsement Key is an asymmetric key pair that is used as proof that a SLB96xx is genuine. The Endorsement Key pair is generated and encrypted outside the SLB96xx in a secure environment by the manufacturer Infineon Technologies AG and then loaded encrypted into the SLB96xx during the production phase. The Endorsement Key is transitively bound to the Platform via the SLB96xx as follows:

- An Endorsement Key is bound to one and only one SLB96xx (i.e., that is a one to one correspondence between an Endorsement Key and a SLB96xx)
- A SLB96xx is bound to one and only one Platform, (i.e., there is a one to one correspondence between a SLB96xx and a Platform.)
- Therefore, an Endorsement Key is bound to a Platform, (i.e., there is a one to one correspondence between an Endorsement Key and a Platform)

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

The Direct Anonymous Attestation Protocol (DAA) provides evidence that the TPM holds a valid Attestation Identity Key without revealing the attestation information.

Support for the Root of Trust for Measurement

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

Generation and import of the Endorsement Key pair

The Endorsement Key (EK) pair stored in the TPM is generated outside the TPM with the TPM Personalization Certification Authority (TPM-CA) located within the secure production area of the TOE. The EK pair and the associated EK credential are loaded into the TPM during the manufacturing process at the TOE lifecycle phase “Manufacturing”. The TPM-CA is located at the production site of the TOE in a secure room. The TPM-CA consists of a Personalization Dataset Generator (PDG) including a hardware security model (HSM-PDA), a Certification Authority (INCA) and a database server. The HSM-PDG generates the EK pair and encrypts the key pair using a master key and the algorithm three key Triple-DES in CBC mode. The encrypted EK pair is then stored in the database server. The INCA creates the EK credential by certifying the public part of the EK. The EK credential is also stored at the database server. For the production process the plain EK pair is encrypted with a TPM individual transport key and transported to the production facility. The personalization process generates the TPM individual transport key and loads this key together with the encrypted EK pair and the EK credential into the TPM. Within the TPM the EK pair is encrypted with the transport key and stored in a shielded location. The generation and import process of the Endorsement Key pair is done completely in the secure production area of the TOE.

To simplify system integration into existing PC mainboards, the SLB96xx uses the LPC interface (Low Pin Count) as defined by Intel.

With these capabilities, the SLB96xx is able to realize the issue of the TPM Main Specification to insert a trusted subsystem – called the “root of trust” – into the PC platform, which is able to extend its trust to other parts of the whole platform by building a “chain of trust”, where each link extends its trust to the next one. As a result, the TPM extends its trustworthiness, providing a Trusted PC for secure transactions. As an example the TPM is able to calculate hash-values of the BIOS at boot time as integrity metric. Once this metric is available, it is saved in a secure memory location. Optionally, it could be compared to some predefined values and the boot process could be aborted on mismatch.

During the boot process, other integrity metrics are collected from the platform, e.g. the boot loader and the operating system itself. Device drivers may be hashed and even hardware like PCI cards can be detected and identified. Every metric obtained is concatenated to the already available metrics. This gives a final metric, which describes the operational state of the whole platform and the state of its system integrity.

A challenger may now ask the platform for these metrics and make informed decisions on whether to trust it based on the metric values obtained. To support the privacy issue, the user of the platform may restrict the SLB96xx in answering to any challenge, but the user is never able to make the SLB96xx report false metrics. Moreover, the user is able to create several identities for his interactions.

Offering these features to a system, the SLB96xx can be used in a wide field of applications, e.g. in a remote access network to authenticate platforms to a server and vice versa. Concerning e-commerce transactions, contracts can be signed with digital signatures using the SLB96xxasymmetric encryption functionality. Regarding a network scenario, the client PCs equipped with a SLB96xx are able to report their platform status to the server so that the network administration is aware of their trustworthiness. In conclusion, the SLB96xx acting as a service provider to a system helps to make transactions more secure and trustworthy.

The Target of Evaluation (TOE), the SLB96xx, consists of the following hardware and firmware components.

The hardware of the SLB96xx is based on the SLE70-Family architecture with additional components and is manufactured by the Infineon Technologies AG.

The IC, whose block diagram is shown in Figure 1, consists of a dedicated microprocessor (CPU) with a MMU (Memory Management Unit), several different memories, security logic, shield, a timer, an interrupt-controlled I/O interface and a RNG (**R**andom **N**umber **G**enerator) are integrated on the chip. Additionally, a hardware hash accelerator and a specialized interface the Low Pin Count interface (LPC) have been added. This LPC interface is the main interface of the chip.

The CPU is a real 16-bit CPU-architecture and is compatible to the Intel 80251 architecture. The major components of the core system is the CPU (Central Processing Unit), the MMU (Memory Management Unit) and MED (Memory Encryption/Decryption Unit). The CPU control each other in order to detect faults and serve by this for data integrity. The TOE implements a full 16 MByte linear addressable memory space for each privilege level, a simple scalable Memory Management concept and a scalable stack size. The flexible memory concept consists of ROM- and Flash-memory as part of the non volatile memory (NVM), respectively EEPROM. For the EEPROM memory the Unified Channel Programming (UCP) memory technology is used.

The SLB96xx uses an external clock of 33 MHz where is compliant to the definition of the LPC interface. The PLL unit allows operating the core controller of the SLB96xx with a multiplication factor over the divided external clock signal or free running with maximum frequency. The checksum module allows simple calculation of checksums per ISO 3309 (16 bit CRC).

Three modules for cryptographic operations are implemented on the TOE. The two cryptographic co-processors serve the need of modern cryptography: The symmetric co-processor (SCP) for AES hardware acceleration. The Asymmetric Crypto Co-processor, called Crypto2304T in the following, is used for RSA-2048 bit (4096-bit with CRT). The third module named HASH provides the Secure Hash Algorithm-1 (SHA-1).

To sum up, the TOE is a powerful security IC with a large amount of memory and special peripheral devices with both improved performance and optimized power consumption at minimal chip size.

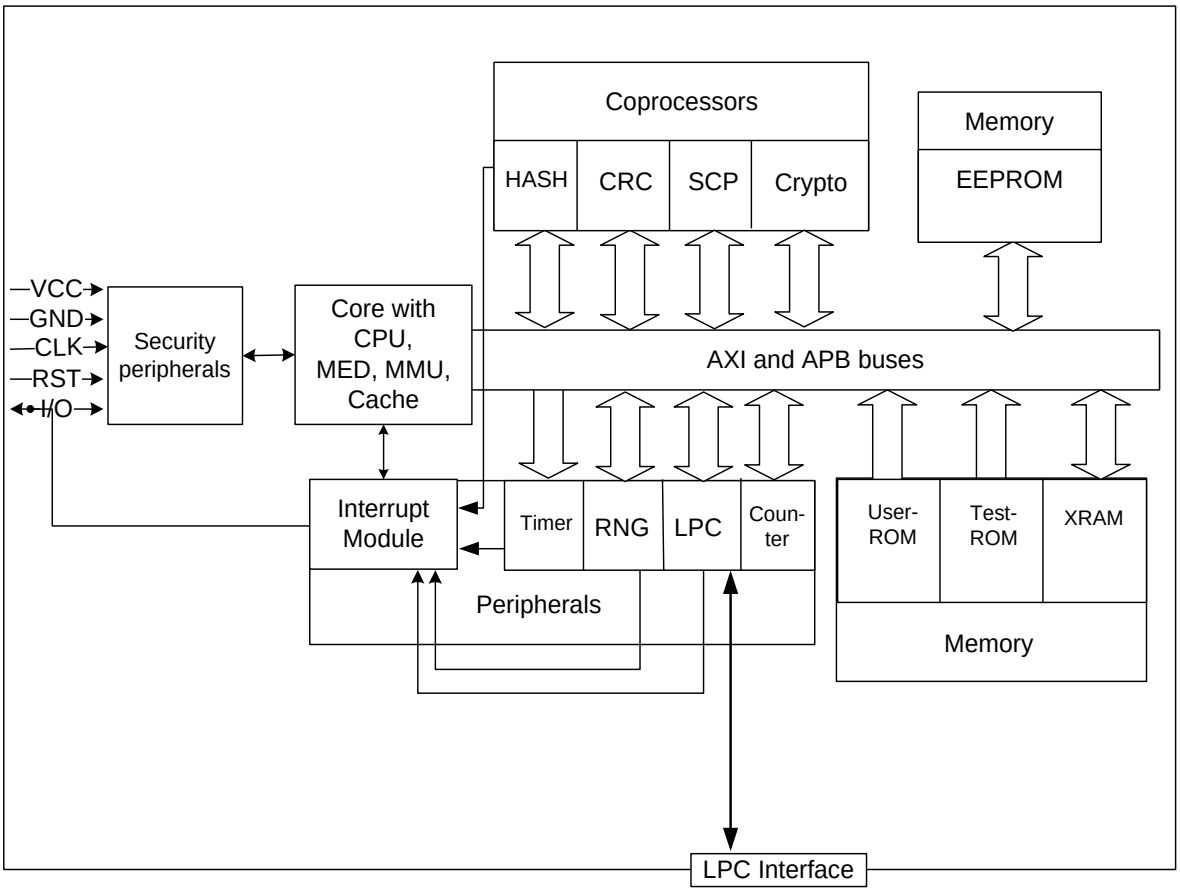


Figure 1: Block diagram of the SLB96xx

The firmware required for operating the chip includes an operating system that provides the TCG functionality specified in the TPM Main Specification. The chip initialisation routine with security checks and identification mode as well as test routines for production testing are located in a separate test ROM. The firmware also provides the mechanism for updating the protected capabilities once the TOE is in the field as defined in the TPM_FieldUpgrade command of the TPM Main Specification. The field upgrade can only be downloaded to the chip if it has been encrypted and signed by the manufacturer Infineon Technologies AG. The Figure 2 shows the firmware block diagram of the SLB96xx.

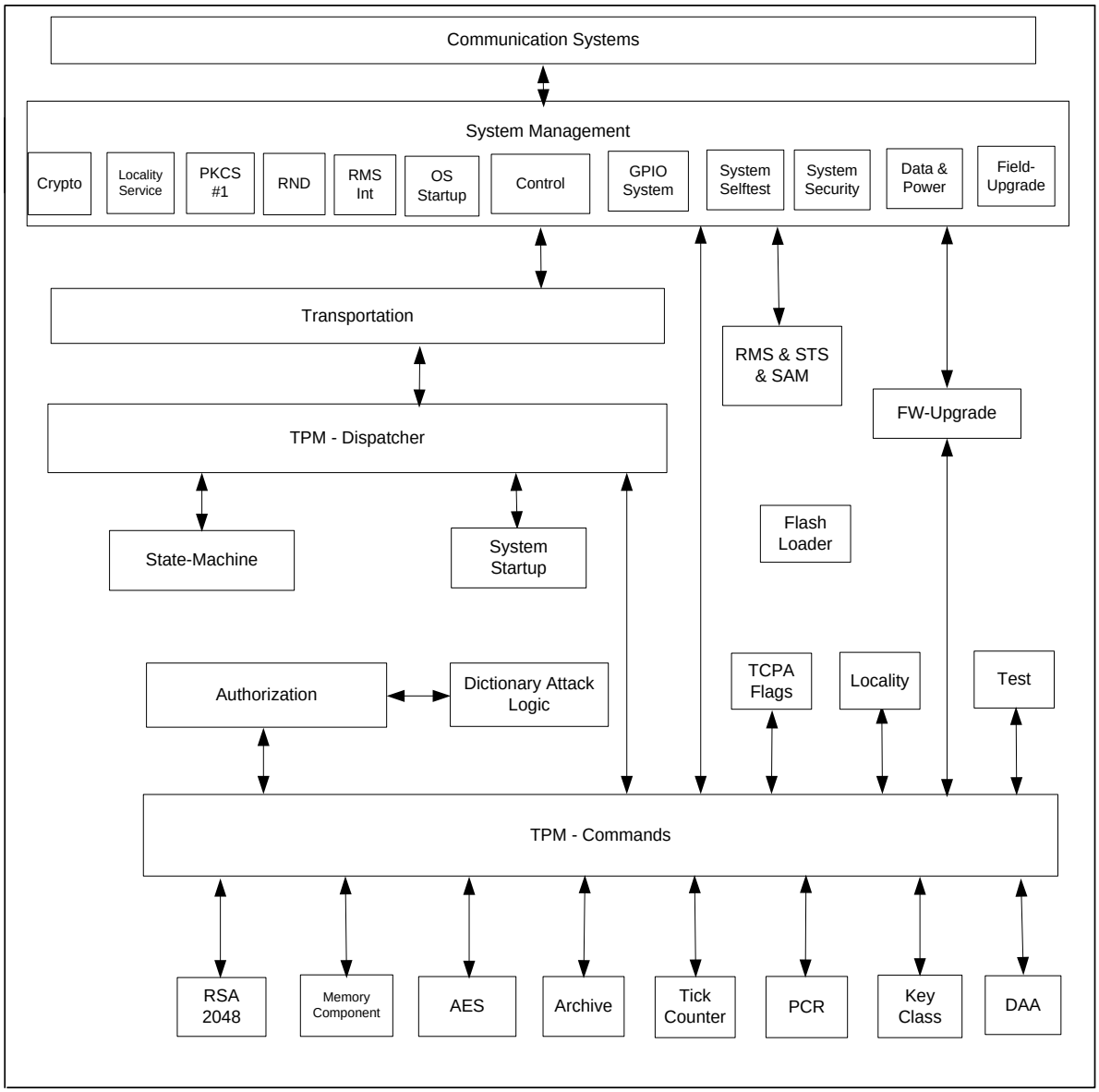


Figure 2: Firmware block diagram of the SLB96xx

2.2 Scope of the TOE

The TOE manufactured by Infineon Technologies AG, comprises the *hardware* of the security controller, type SLB96xx, and the associated *firmware* required for operation provided in ROM and EEPROM

2.2.1 Hardware of the TOE

The *hardware part* of the TOE (cf. Figure 1) as defined in the PP [8] is comprised of:

- Security Peripherals (filters, sensors)
- Core System
  - with proprietary CPU implementation of the Intel MCS251 standard architecture from functional perspective
  - Cache with post failure detection
  - Memory Encryption/Decryption Unit (MED)
  - Memory Management Unit (MMU)
- Memories
  - Read-Only Memory (ROM)
  - Random Access Memory (RAM)
  - EEPROM Flash memory
- Coprocessors
  - Crypto2304T for asymmetric algorithms like RSA
  - Symmetric Crypto Co-processor AES standard (SCP)
  - Hash accelerator (HASH) for the algorithm SHA-1
  - Checksum module (CRC)
- Random number generator (RNG)
- Interrupt module (INT)
- Timer (TIM)
- Buses (BUS)
  - AXI™ Memory Bus
  - APB Peripheral B
- Low Pin Count interface (LPC)
- Tick Counter

2.2.2 Firmware of the TOE

The entire firmware of the TOE consists of different parts. The one part is the operating system which includes the TPM application, the System Management and the Endorsement Key and is used to operate the IC. The operating system includes also the capability for updating the protected capabilities once the TOE is in the field (TPM_FieldUpgrade). Note that it is possible to update e.g. the TPM firmware version v04.32.0879 to a new certified firmware version e.g. v04.43.0258.00.
The other firmware part is the Self Test Software (STS), the Service Algorithm Minimal (SAM), the Resource Management System (RMS) and the Flash Loader. The STS routines are stored in the especially protected test ROM and are not accessible for the user software (application).
The entire operating system of the TOE (cf. Figure 2) as defined in the PP [8] is comprised of:

- Communication System
- System Management including:
  - GPIO System
  - Crypto
  - Locality System
  - PKCS#1
  - RND (DRNG)
  - RMS Int
  - OS Startup
  - Control
  - System Selftest
  - Security System
  - Data & Power
  - Field Upgrade
- Transportation
- TPM-Dispatcher
- System Startup
- State-Machine
- Authorization
- Dictionary Attack Logic
- TPM-Command
- TcpaFlags
- Locality
- Test
- RSA2048
- Memory Components
- AES
- Archive
- Tick Counter
- PCR
- Key Class

- FW-Upgrade
- Flash Loader
- DAA

2.2.3 Guidance documentation

The guidance documentation consists of a set of information containing the description of all interfaces to operate the TOE. The list of the guidance documentation is given in Table 1, section Guidance Documentation.

2.2.4 Forms of delivery

The TOE is delivered in form of complete chips which include the hardware, the firmware (operating system), the Endorsement Key Pair, and the guidance documentation. The TOE is finished and the extended test features are removed. The TOE is delivered in different packages (e.g. TSSOP and VQFN) which are listed in the document [88].

2.2.5 Production sites

The TOE silicon is produced in the production site Dresden. The Chip Marking includes an assembly side code which defines the assembly site. The exact coding of the chip marking is described in [10], section 5.3 or 6.3 “Chip Marking”.

Table 2: Production site in chip identification

Production Site	Chip Identification
Dresden, Germany	byte number 13 (Fab number): 02_H

The delivery measures are described in the ALC_DVS aspect.

2.2.6 Life cycle of the TOE

The life cycle of the TOE as part of the evaluation covers phase 1 “Development” and phase 2 “Manufacturing” as defined in the PP [8] section 1.3.3. The phase 1 includes the TPM development, the phase 2 includes the TPM manufacturing, the TPM conformance testing, the TPM-Mfg EK key pair download and the TPM-Mfg EK credential issuance.

3 Conformance Claims

3.1 CC Conformance Claim

This Security Target (ST) and the TOE claim conformance to Common Criteria version v3.1 part 1 [2], part 2 [3] and part 3 [4].

Conformance of this ST is claimed for:
Common Criteria part 2 extended and Common Criteria part 3 conformant.

3.2 PP Claim

This Security Target is in **strict conformance** to the
Trusted Computing Group Protection Profile PC Client Specific Trusted Platform Module Family 1.2; Level 2, Revision 116 [8].
The Protection Profile (PP) [8] is registered and certified by the Bundesamt für Sicherheit in der Informationstechnik[1] (BSI) under the reference BSI-PP-0030-2008-MA-01, Version: 1.2, dated 18.05.2011.

The security assurance requirements of the TOE are according to the Security IC Platform Protection Profile [8]. They are all drawn from Part 3 of the Common Criteria version v3.1.

3.3 Package Claim

This Security Target does not claim conformance to a package of the PP [8].

The assurance level for the TOE is EAL4 augmented with ALC_FLR.1 and AVA_VAN.4 defined in CC part 3 [3].

3.4 Conformance Claim Rationale

This security target claims strict conformance only to one PP, the PP [8].

The Target of Evaluation (TOE) is a complete solution implementing the TCG Trusted Platform Module Main, Specification Version 1.2, [5] [6] [7] and the TCG PC Client Specific TPM Interface Specification, Version 1.2 Final, Revision 1.00 [9] as defined in the PP [8] section 1.3.1, so the TOE is consistent with the TOE type in the PP [8].

The security problem definition of this security target are consistent with the statement of the security problem definition in the PP [8], as the security target claimed strict conformance to the PP [8] and no other threats, organisational security policies and assumptions are added.

The security objectives of this security target are consistent with the statement of the security objectives in the PP [8], as the security target claimed strict conformance to the PP [8] and no other security objectives are added.

The security requirements of this security target are consistent with the statement of the security requirements in the PP [8], as the security target claimed strict conformance to the PP [8]. All assignments and selections of the security functional requirements are done in the PP [8] and in this security target at section 7.1, e.g. the security functional requirement FCS_RNG.1 “Generation of Random Numbers “.

[1] BSI is the German abbreviation for Federal Authority for Information Security

3.5 Application Notes

The functional requirement FCS_RNG.1 is a refinement of the FCS_RNG.1 defined in the Protection Profile [8] according to “Anwendungshinweise und Interpretationen zum Schema (AIS),” [11].

4 Security Problem Definition

The content of the PP [8] applies to this chapter completely.

4.1 Threats

The threats are directed against the assets.

The threats to security are defined in the PP [8] section 4.1, no other threats are added.

The primary assets concern the TSF and the User Data that includes the data as well as program code (Embedded Firmware). These assets have to be protected while being executed as well as when the TOE is not in operation. This leads to the following primary assets:

- Embedded Firmware
- User Data
- TSF Data
- Hardware of TOE

4.2 Organisational Security Policies

The organisational security policies are defined in the PP [8] section 4.2, no other organisational security policies are added.

4.3 Assumptions

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

The TOE assumptions to the IT environment are defined in the PP [8] section 4.3, no other assumptions are added.

The TCG subsystem, in which the TPM is used, is a trusted subsystem that is an integral part of a computing platform that consists of logical components including the TPM, the Connection module (PCCON) and the Trusted Platform Support Services (TSS).

In general the TPM provides cryptographic capabilities and protected storage.

The Connection module (PCCON) provides the connection to the computing platform and the Root of Management Trust (RMT). The TPM relies on the PCCON module for all communication with the platform and for the RMT.

The TSS is a set of functions and data that are common to all types of platforms, which are not required to be trustworthy and therefore do not need to be part of the TPM.

5 Security Objectives

This section shows the security objectives which are relevant for the TOE. For this section the PP [8] can be applied completely.

5.1 Security Objectives for the TOE

The security objectives of the TOE are defined and described in the PP [8] section 5.1, no other security objectives are added.

5.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are described in the PP [8] section 5.2, no other security objectives for the operational environment are added.

5.3 Security Objectives Rationale

The security objectives rationale is described in the PP [8] section 5.3. No other security objectives rationale are added.

6 Extended Components Definition

The extended component “FCS_RNG Generation of random numbers” (FCS_RNG.1) is defined in the following sections.

Family “ Generation of Random Number (FCS_RNG)”

The family “Generation of Random Numbers (FCS_RNG.1)” has to be newly created according the new version of the “Anwendungshinweise und Interpretationen zum Schema (AIS)” [11]. This security functional component is used instead of the functional component FCS_RNG.1 defined in the protection profile [8].

The family “Generation of Random Numbers (FCS_RNG.1)” is specified as follows (Common Criteria Part 2 extended).

FCS_RNG Generation of random numbers

Family behavior

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

Component leveling:

FCS_RNG Generation of random numbers	1

FCS_RNG.1	Generation of random numbers, requires that the random number generator implements defined security capabilities and that the 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, non-physical true, deterministic, hybrid physical, hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].
FCS_RNG.1.2	The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

Application Note 1: The functional requirement FCS_RNG.1 is a refinement of the FCS_RNG.1 defined in the Protection Profile [8] according to “Anwendungshinweise und Interpretationen zum Schema (AIS)” [11].

7 IT Security Requirements

For this section the PP [8] section 6 can be applied completely.

7.1 Security Functional Requirements for the TOE

The security functional requirements (SFR) for the TOE are defined and described in the PP [8] section 6.1 and in section 7.1.1 for the extended security requirement FCS_RNG1.
All assignments and selections of the security functional requirements are done in the PP [8] with the exception of the following SFRs. The operations completed in the ST are marked in *italic* font.

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.
Dependencies: No dependencies

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

(1) Management of the TPM modes of operation,

(2) Management of Delegation Tables and Family Tables,

(3) Management of security attributes of keys,

(4) Management of security attributes of PCR,

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

(6) Management of security attributes of monotonic counters,

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

(8) *None.*

FPT_TDC.1 Inter-TSF basic TSF data consistency

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

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret *authentication reference data of the User using operatorAuth, TPM owner, delegated entities, owner of entities, user of entities* when shared between the TSF and another trusted IT product.

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

FCS_CKM.1/AES Cryptographic key generation

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

FCS_CKM.1.1/AES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *AES key generator* and specified cryptographic key sizes *128 bits* that meet the following: *the AES key is a 128 bit random number.*

FCS_CKM.4 Cryptographic key destruction

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

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method: *key destruction* that meets the following: *FIPS PUB 140-1 [FIPS_140], section 4.8.5 (overwriting all bits with "1" or with a random value).*

FCS_COP.1/RSA_Sig Cryptographic operation

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

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

Signature Generation (with or without CRT):

According to section 5.2.1 RSASP1 in PKCS v2.0 RFC3447,
for u = 2, i.e., without any (r_i, d_i, t_i), i >2,
therefore without 5.2.1.2.b (ii)&(v).
Without 5.2.1.1.
5.2.1.2.a, only supported up to $n < 2^{2048}$

Signature Verification:

According to section 5.2.2 RSAVP1 in PKCS v2.0 RFC3447,
without 5.2.2.1.

FCS_COP.1/RSA_Enc Cryptographic operation

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

FCS_COP.1.1/RSA_Enc

The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm RSA encryption scheme] [5] *section 31.1.1, 31.1.2 and 31.1.3* and cryptographic key sizes RSA 512, 1024, 2048 that meet the following: PKCS#1 V2.0 [PKCS].

Encryption:

According to section "5.1.1 RSAEP" in PKCS v2.1 RFC3447.
Without 5.1.1.1.

Decryption (with or without CRT):

According to section "5.1.2 RSADP" in PKCS v2.1 RFC3447,
for u=2, i.e., without any (r_i, d_i, t_i), i > 2,
therefore without 5.1.2.2.b (ii)&(v).
Without 5.1.2.1.
5.1.2.2.a, only supported up to to $n < 2^{2048}$.

FCS_COP.1/SymEnc2 Cryptographic operation

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

FCS_COP.1.1/SymEnc2

The TSF shall perform *symmetric encryption and decryption* in accordance with a specified cryptographic algorithm
- AES, mode CBC and mode CTR
and cryptographic key sizes
- 128 bit for AES,
that meet the following:
- U.S. Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory (ITL),
Advanced Encryption Standard (AES), FIPS PUB 197; [FIPS_197],

and NIST Special Publication 800-38A, 2001 Edition, Recommendation for Block Cipher Modes of Operation Methods and Techniques [N800]

FIA_UID.1 Timing of identification

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

FIA_UID.1.1 The TSF shall allow

(1) to execute commands indicated in the PP [8], Table 12 column RQU as not requesting authentication,

(2) accessing objects where entity owner has given the user “World” access based on the value of TPM_AUTH_DATA_USAGE,

(3) *to execute the commands TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END*

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

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

FIA_UAU.1 Timing of authentication

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

FIA_UAU.1.1 The TSF shall allow

(1) to execute commands indicated in the PP [8] Table 12 column RQU as not requesting authentication,

(2) accessing objects where entity owner has given the user “World” access based on the value of TPM_AUTH_DATA_USAGE,

(3) *to execute the commands TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END,*

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

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

FIA_UAU.5 Multiple authentication mechanisms

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

FIA_UAU.5.1 The TSF shall provide

(1) OIAP authorization session,

(2) OSAP authorization session,

(3) DSAP authorization session,

(4) Transport session,

(5) Commands which require authorization and are executed outside a authorization session

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the *rule “dictionary attack”. The dictionary attack is a mechanism to manage authentication failure handling to mitigate dictionary attacks. The dictionary attack provides two different objects, a single object mode, which means that all detected authentication failure are collected at the same counter, and the full object mode, which means that the detected authentication error could be distinguished and collected on different counters. The maximal number of authentication errors is 11, this default*

value can be reduced by the user software. If the dictionary attack mechanism has the maximum number of authentication errors detected, the Activation Flag is set to TRUE and two different ways of processing could be started. For the first way logic is activated to add a delay time into the processing before the next command could be processed. For each detected authentication error the delay time duration is doubled until a maximum value. For the second way the same procedure as for the first way is started, additionally the TOE is deactivated after a predefined number of detected authentication failures. The reset of the failure counter and delay timer is done with a correct authentication process, after predefined time duration or by the TPM owner. The activation of the TPM could be done only by the TPM owner.

FIA_AFL.1 Authentication failure handling

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

FIA_AFL.1.1 The TSF shall detect when *ten* unsuccessful authentication attempts occur related to authentication attempts for the same user.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall

(1) Set the Action Flag to TRUE,

(2) Two different ways of processing could be started. For the first way logic is activated to add a delay time into the processing before the next command could be processed. For each detected authentication error the delay time duration is doubled until a maximum value. For the second way the same procedure as for the first way is started, additionally the TOE is deactivated after a predefined number of detected authentication failures.

FDP_ACF.1/Deleg Security attribute based access control

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

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

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

(1) The TSF shall disallow the execution of a command in a DSAP session if the permission of this command is not set in the delegation table row in the Delegation owner blob used for the DSAP session,

(2) The TSF shall disallow the execution of a command in a DSAP session if the PCR_SELECTION of the DSAP session is not NULL and the pcrInfo of the

DSAP session does not match the current PCR value of the PCR_SELECTION and locality.

FDP_ACF.1.3/Deleg

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules*: if the TPM command is listed in the table at [6], section 20.2.1 "Owner Permission Settings" or the key is listed in the table at [6] section 20.2.3 "Key Permission Settings", then the TPM owner or the key user can delegate that capability to a trusted process*.

FDP_ACF.1.4/Deleg

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *if the TPM command is listed in the table at [6], section 20.2.2 "Owner commands not delegated" or the key is listed in the table at [6], section 20.2.4 "Key commands not delegated", then the command can not be delegated***.**

FDP_ACF.1/KeyMan Security attribute based access control

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

FDP_ACF.1.1/KeyMan

The TSF shall enforce the Key Management SFP to objects based on the following:

(1) subjects: commands with security attributes ownerAuth, srkAuth, AuthData,locality, physical presence;

(2) objects:

(a) EK with the SFR-related security attribute ownership of the TOE,

(b) SRK with the SRF related security attributes disableOwnerClear and disableForceClear of the TOE,

(c) User keys with the security attributesauth<DataUsage, keyUsage, keyFlags, and ownerEvict,

(d) Wrapped Key Blob with the security attributes keyUsage, keyFlags, algorithmParms and pcrInfo.

FDP_ACF.1.2/KeyMan

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

(1) The user "World" is allowed to create an EK if the EK does not exist already.

(2) The user "World" is allowed to read the public part of an EK if the TOE is unowned.

(3) The TPM owner is allowed to read the public part of an EK.

(4) The user "World" is allowed to create an SRK if the ownership flag is TRUE.

(5) The TPM owner is allowed to delete an SRK if the disableOwnerClear flag is FALSE.

(6) The user "World" under physical presence is allowed to delete an SRK if the disableForceClear flag is FALSE.

(7) The user authenticated as TPM owner and the owner of the SRK is allowed to generate an AIK.

(8) The TPM owner is allowed to activate the AIK if the imported blob is a TPM_EK_BLOB structure and the actual state meets the identified PCR values and the locality.

(9) The TPM owner is allowed to use the AIK for signing audit data, quoted data, or a tick stamped blob.

(10) The entity owner of a key with the security attribute keyUsage, TPM_KEY_STORAGE=TRUE, is allowed to generate an User Key and export this User key wrapped with the key the owns key except this entity owner is not the TPM owner and the key to generated is an AIK

(11) The Entity owner of the key to be used for import of Wrapped Key Blob is allowed to import a User key in a Wrapped Key Blob if the security attribute keyUsage, TPM_KEY_STORAGE=TRUE.

(12) The entity owner is not allowed to use a User key if at least one of the following conditions is met:

(a) the security attribute authDataUsage of the User Key object for access does not match the authentication status of the subject,

(b) the security attribute usageAuth of the User Key object for access does not match the authentication data used by the user bound to the subject,

(c) the security attributes keyUsage or algorithmParms or keyFlags of the User Key object does not allow to use the command to be executed,

(d) the security attribute PCRInfo of the User Key object does not allow to use the object in the actual state of the identified PCR and locality

(13) The TPM owner is allowed to delete a User key if the security attribute OwnerEvict, OwnerEvict=FALSE.

FDP_ACF.1.3/KeyMan

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:
(a) The execution of the commands TPM_CreateWrapKey, TPM_LoadKey, TPM_LoadKey2, TPM_Unseal, TPM_GetPubKey, TPM_CertifyKey, TPM_CertifyKey2, TPM_MakeIdentity, TPM_DSAP, TPM_ChangeAuthAsymStart, TPM_CMK_CreateKey, TPM_CMK_SetRestrictions, TPM_CMK_CreateBlob and TPM_CMK_ConvertMigration depends on the values of the security attribute TPM_KEY_FLAG (keyFlags).
(b) The execution of the commands TPM_CMK_SetRestrictions, TPM_ChangeAuthAsymStart, TPM_Take_Ownership, TPM_Seal, TPM_Unseal, TPM_Sealx, TPM_Unbind, TPM_Sign, TPM_CertifyKey, TPM_LoadKey, TPM_LoadKey2, TPM_CreateWrapKey, TPM_MakeIdentity, TPM_GetPubKey, TPM_MigrateKey, TPM_DSAP, TPM_Quote, TPM_ActivateIdentity, TPM_ConvertMigrationBlob, TPM_CertifySelfTest, TPM_CMK_CreateKey, TPM_CMK_ConvertMigration, TPM_TickStampBlob and TPM_EstablishTransport depends on the values of the security attribute TPM_KEY_USAGE (KeyUsage).

(c) The execution of the commands TPM_Startup, TPM_KeyControlOwner, TPM_FlushSpecific and TPM_EvictKey depends on the values of the security attribute OwnerEvict.

(d) The execution of the commands TPM_TakeOwnership, TPM_AuthorizeMigrationKey, TPM_CMK_CreateTicket and TPM_CMK_CreateBlob depends on the values of the security attribute algorithmParams.
(e) The execution of the commands TPM_Seal, TPM_Unseal, TPM_LoadKey, TPM_LoadKey2, TPM_MakeIdentity, TPM_GetPubKey, TPM_Sealx, TPM_CertifyKey, TPM_CertifyKey2, TPM_CMK_CreateKey, TPM_NV_WriteValue, TPM_NV_WriteValueAuth, and TPM_NV_ReadValueAuth depends on the values of the security attribute pcrInfo .
(f) The read of the public portion of the key PUBEK is allowed without owner authorization if the readPUBEK (flag TPM_EF_READPUBEK in TPM_PERMANENT_FLAGS) is set to TRUE.

FDP_ACF.1.4/KeyMan
The TSF shall explicitly deny access of subjects to objects based on the following additional rules:
(a) *The read of the public portion of the key PUBEK is not allowed if the readPUBEK (flag TPM_EF_READPUBEK in TPM_PERMANENT_FLAGS) is set to FALSE.*

FMT_MSA.3/KeyMan Static attribute initialization

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

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

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

FDP_ACF.1/MigK Security attribute based access control

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

FDP_ACF.1.1/MigK
The TSF shall enforce the Key Migration SFP to objects based on the following:
(1) Subjects: TPM owner, Entity owner of the key with security attributes restrictDelegate and migrationSheme,
(2) Objects:
  (a) User key with security attribute migratable,
  (b) Wrapped Key Blob with the security attribute payload type,
  (c) Migration Key Blob with the security attribute payload type,
  (d) Certified Migration Key Blob with the security attributes payload type and migrationAuth.

FDP_ACF.1.2/MigK

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

(1) The Entity owner of a migratable User key is allowed to create a Wrapped Key Blob for this migratable key by means of the command TPM_CMK_CreateKey, if it is authorized for use of the CMK Migration Approval Ticket and in case of delegated commands the restrictions for the migration of keys are fulfilled.

(2) The Entity owner of a migratable User key authorized for use of the Migration key authorization ticket is allowed to create a Migration Key Blob for this migratable key by means of the command TPM_CreateMigrationBlob.

(3) The Entity owner of a certifiable migratable User key authorized for use of the Migration key authorization ticket and the Restriction Ticket is allowed to create a Certified Migration Key Blob for this migratable key by means of the command TPM_CMK_CreateBlob.

(4) The Entity owner of private part of the migration User key is allowed to migrate a Migration Key Blob and a Certified Migration Key Blob to a conversion key by means of the command TPM_MigrateKey,

(5) The Entity owner of the private part of migration User key is allowed to convert a Migration Key Blob by means of the command TPM_ConvertMigrationBlob and a Certified Migration Key Blob by means of the command TPM_CMK_ConvertMigration if in case of delegated commands the restrictions for the migration of keys are fulfilled.

FDP_ACF.1.3/MigK

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:
(a) *The execution of the commands TPM_CreateMigrationBlob, TPM_ConvertMigrationBlob, TPM_CMK_CreateKey, TPM_CMK_CreateBlob and TPM_CMK_ConvertMigration depends on the value of the security attribute payload type.*
(b) *The execution of the commands TPM_CreateMigrationBlob, and TPM_CMK_CreateBlob depends on the value of the security attribute migrationKeyAuth.*
(c) *The execution of the command TPM_CMK_CreateKey depends on the value of the security attribute migrationAuthorityApproval.*

FDP_ACF.1.4/Mig

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

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

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

FDP_ACF.1.1/M&R

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

(1) Subjects:
(a) SHA-1 session,
(b) user with the security attribute locality,
(c) entity owner of the signature key with the security attribute usageAuth,,
(2) Objects:
(a) PCR with the security attribute pcrReset, pcrResetLocal, pcrExtend-Local
(b) Signature key with the security attribute User Key.

FDP_ACF.1.2/M&R

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

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

(2) The SHA-1 session is allowed to calculate the new digest of the SHA-1 session as SHA-1 hash value of the digest of the SHA-1 session and the presented data by command TPM_SHA1Update.

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

(4) The SHA-1 session is allowed (i) to finish the calculation of the digest of the SHA-1 session as SHA-1 hash value of the digest of the SHA-1 session and the presented data and (ii) to extend the value of the indicated PCR by command TPM_SHA1CompleteExtend.

(5) If the pcrReset is TRUE the command TPM_Startup is allowed to set a PCR to 0xFF…FF.

(6) If the pcrReset is FALSE the command TPM_Startup is allowed to set a PCR to 0x00…00.

(7) If the user presents the locality matching the security attribute pcrResetLocal of the selected PCR and the pcrReset of this PCR is TRUE, than the command TPM_PCR_Reset is allowed to reset this PCR to 0x00…00 or 0xFF…FF, where the concrete value is defined in the platform specific specification of the TOE.

(8) If the user presents the locality matching the security attribute pcrExtendLocal of the selected PCR the command TPM_SHA1CompleteExtend is allowed (i) to finish the calculation of the digest of the SHA-1 session as SHA-1 hash value of the digest of the SHA-1 session and the presented data and (ii) to extend the value of the selected PCR with the final digest of the SHA-1 session.

(9) If the user presents the locality matching the security attribute pcrExtendLocal of the selected PCR the command TPM_Extend is allowed to extend the value of the selected PCR with the presented data.

(10) The user “World” is allowed to read the PCR object with the command TPM_PCRRead.

(11) The entity owner is allowed to quote the PCR indicated by the parameter targetPCR with the User key, which security attribute keyUsage equals to TPM_KEY_SIGNING, TPM_KEY_IDENTITY, or TPM_KEY_LEGACY, by means of the command TPM_Quote or TPM_Quote2.

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

(13) Additional rules for operations, based on security attributes of the subjects and objects: *none.*

FDP_ACF.1.3/M&R

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:
(a) *The execution of the command TPM_PCR_Reset depends on the values of the security attributes pcrReset and pcrResetLocal.*
(b) *The execution of the commands TPM_SHA1CompleteExtend and TPM_Extend depends on the value of the security attribute pcrExtendLocal.*
(c) *The execution of the commands TPM_Quote and TPM_Quote2 depends on the value of the security attribute KeyUsage.*

FDP_ACF.1.4/M&R

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

FMT_MSA.3/M&R Static attribute initialization

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

FMT_MSA.3.1/M&R

The TSF shall enforce the Measurement and Reporting SFP to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/M&R

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

FDP_ACF.1/NVS Security attribute based access control

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

FDP_ACF.1.1/NVS

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

(1) Subjects: user “World”, entity owner and TPM owner with the security attributes physical presence and current PCR values,

(2) Objects: NV storage with the security attributes nvLocked, noOwnerNVWrite, pcrInfoRead, pcrInfoWrite, localityAtRelease, and permissions TPM_NV_PER_READ_STCLEAR, TPM_NV_PER_WRITE_STCLEAR TPM_NV_PER_AUTHWRITE, TPM_NV_PER_OWNERWRITE TPM_NV_PER_PPWRITE, TPM_NV_PER_AUTHREAD, TPM_NV_PER_PPREAD, TPM_NV_PER_OWNERREAD, TPM_MAX_NV_WRITE_NOOWNER.

FDP_ACF.1.2/NVS

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

(1) The user “World” under physical presence is allowed to create NV storage by means of the command TPM_NV_DefineSpace if nvLocked is 0 and noOwnerNVWrite does not exceed TPM_MAX_NV_WRITE_NOOWNER.

(2) The TPM owner is allowed to create a NV storage area by means of the command TPM_NV_DefineSpace.

(3) The user “World” is allowed to write the NV storage area if nvLocked of the TPM_PERMANENT_FLAGS is FALSE and max NV writes without an owner is not exceeded.

(4) The TPM owner is allowed to write an NV storage area by means of the command TPM_NV_WriteValue if

(a) TPM_NV_PER_OWNERWRITE is TRUE,

(b) the user match the requirement for physical presence defined in TPM_NV_PER_PPWRITE,

(c) the locality of the user mach the localityAtRelease defined for the TPM_NV_DATA_AREA and

(d) if pcrInfWrite defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoWrite.

(5) The entity owner is allowed to write an NV storage area by means of the command TPM_NV_WriteValueAuth if

(a) TPM_NV_PER_AUTHWRITE is TRUE,

(b) the user match the requirement for physical presence defined in TPM_NV_PER_PPWRITE,

(c) the locality of the user matches the localityAtRelease defined for the TPM_NV_DATA_AREA and

(d) if pcrInfWrite defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoWrite.

(6) The TPM owner is allowed to read an NV storage area by means of the command TPM_NV_ReadValue if

(a) TPM_NV_PER_OWNERREAD is TRUE,

(b) the user match the requirement for physical presence defined in TPM_NV_PER_PPREAD,

(c) the locality of the user matches the localityAtRelease defined in the pcrInfoRead and

(d) if pcrInfoRead defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoRead.

(7) The Entity owner is allowed to read an NV storage area by means of the command TPM_NV_ReadValueAuth if

(a) TPM_NV_PER_AUTHREAD is TRUE,

(b) the user matches the requirement for physical presence defined in TPM_NV_PER_PPREAD,

(c) the locality of the user matches the localityAtRelease defined in the pcrInfoRead and

(d) if pcrInfoRead defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoRead.

FDP_ACF.1.3/NVS

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:
(a) *The execution of the command TPM_NV_DefineSpace depends on the values of the security attributes nvLocked, pcrInfoRead, pcrInfoWrite, TPM_MAX_NV_WRITE_NOOWNER, TPM_NV_PER_OWNERWRITE, TPM_NV_PER_AUTHWRITE, TPM_NV_PER_AUTHREAD, TPM_NV_PER_ WRITEDEFINE and TPM_NV PER_PPWRITE.*
(b) The execution of the command TPM_NV_WriteValue depends on the values of the security attributes nvLocked, pcrInfoWrite, localityAtRelease TPM_MAX_NV_WRITE_NOOWNER, TPM_NV_PER_OWNERWRITE, TPM_NV_PER_AUTHWRITE, TPM_NV_PER_PPWRITE and TPM_NV_PER_WITE_STCLEAR.
(c) The execution of the command TPM_NV_WriteValueAuth depends on the values of the security attributes pcrInfoWrite, localityAtRelease TPM_NV_PER_AUTHWRITE, TPM_NV_PER_PPWRITE, TPM_NV_PER_WRITEDEFINE and TPM_NV_PER_WITE_STCLEAR.
(d) The execution of the command TPM_NV_ReadValue depends on the values of the security attributes nvLocked, pcrInfoRead, TPM_NV_PER_AUTHREAD, TPM_NV_PER_OWNERREAD, TPM_NV_PER_PPREAD and TPM_NV_PER_READ_STCLEAR.
(e) The execution of the command TPM_NV_ReadValueAuth depends on the values of the security attributes pcrInfoRead, localityAtRelease, TPM_NV_PER_AUTHREAD, TPM_NV_PER_PPREAD and TPM_NV_PER_READ_STCLEAR.

FDP_ACF.1.4/NVS

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

(1) If TPM_NV_PER_READ_STCLEAR is TRUE the NV storage area can not be read after read with a data size of 0 until successful write or TPM_Startup(ST_Clear).

(2) If TPM_NV_PER_WRITE_STCLEAR is TRUE the NV storage area can not be written after write to the specified index with a data size of 0 until TPM_Startup(ST_Clear).

(3) If TPM_NV_PER_WRITEDEFINE is TRUE the NV storage area can not be written after performing the TPM_NV_DefineSpace command and one successful write.

(4) If TPM_NV_PER_GLOBALLOCK is TRUE the NV storage area can not be written after successful write to index 0 until TPM_Startup(ST_Clear).

(5) The access to the commands
(a) TPM_NV_WriteValue is denied if the security attributes TPM_NV_PER_OWNERWRITE and TPM_NV_PER_AUTHWRITE are both set to TRUE.
(b) TPM_NV_ReadValue is denied if the security attributes TPM_NV_PER_OWNERREAD and TPM_NV_PER_AUTHREAD are both set to TRUE.

FDP_ACF.1/MC Security attribute based access control

Hierarchical to: No other components.

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

FDP_ACF.1.1/MC
The TSF shall enforce the Monotonic Counter SFP to objects based on the following:
(1) Subjects: TPM owner, Entity owner of the monotonic counter object, OSAP session, DSAP session,
(2) Objects: Monotonic counter with security attribute countID.

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

(1) The TPM owner and Delegated entity are allowed to create a Monotonic counter, OSAP and DSAP sessions are required for creation of the Monotonic counter.

(2) The Entity owner of the monotonic counter object is allowed to increment the Monotonic counter if the countID is set in TPM_STCLEAR_DATA for the current boot cycle.

(3) The user “World” is allowed to read the Monotonic counter value if he addresses the Monotonic counter object correctly with valid countID.

(4) The Entity owner of the monotonic counter object is allowed to release the Monotonic counter.

(5) The TPM owner is allowed to release the Monotonic counter.

FDP_ACF.1.3/MC
The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:
(a) *The execution of the commands TPM_IncrementCounter, TPM_ReadCounter, TPMReleaseCounter and TPM_ReleaseCounterOwner depends on the values of the security attribute countID*.

FDP_ACF.1.4/MC

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

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

FDP_ACF.1/EID Security attribute based access control

Hierarchical to: No other components.

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

FDP_ACF.1.1/EID

The TSF shall enforce the Export and Import of Data SFP to objects based on the following:

(1) Subjects: TPM owner with security attribute locality, Entity owner with security attribute locality, user “World”,

(2) Objects:

(a) Sealed data with security attribute pcrInfo and tpmProof,

(b) Context with the security attribute resourceType and tpmProof,

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

FDP_ACF.1.2/EID

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

(1) The Entity owner of the key to be used for export of sealed data is allowed to export Sealed Data if this export key has the security attribute TPM_KEY_STORAGE and is not migratable.

(2) The Entity owner of the key to be used for import of sealed data is allowed to import Sealed Data if

(a) this import key have the security attribute TPM_KEY_STORAGE and is not migratable,

(b) the security attributes pcrInfo of sealed data blob shall match to the values in the PCR indicated by pcrInfo,

(c) the security attributes tmpProof of sealed data blob shall match to the values tpmProof in the TPM_PERMANENT_DATA of the TOE.

(3) The user “World” is allowed to save Context if the resourceType is TPM_RT_KEY, TPM_RT_AUTH, TPM_RT_TRANS or TPM_RT_DAA_TPM.

(4) The user “World” is allowed to load Context if

(a) the resourceType is TPM_RT_KEY, TPM_RT_AUTH, TPM_RT_TRANS or TPM_RT_DAA_TPM and

(b) the tpmProof used as secret for the HMAC of the context match the tpmProof in TPM_PERMANENT_DATA.

(5) The Entity owner of the private part of the bind key is allowed to unbind a Bound blob if the payload type is TPM_PT_BIND.

FDP_ACF.1.3/EID

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

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

FDP_ACF.1.4/EID

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

FMT_MSA.3/EID Static attribute initialization

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

FMT_MSA.3.1/EID

The TSF shall enforce the Export and Import of Data SFP to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/EID

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

FPT_FLS.1 Failure with preservation of secure state

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

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: failure of any crypto operations including RSA encryption, RSA decryption, SHA-1, RNG, RSA signature generation, HMAC generation; failure of any commands or internal operations *and authorization (dictionary attack)*.

FMT_MSA.3/DAA Static attribute initialization

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

FMT_MSA.3.1/DAA

The TSF shall enforce the DAA SFP to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/DAA

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

FPT_PHP.3 Resistance to physical attack

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

FPT_PHP.3.1 The TSF shall resist physical manipulation and physical probing to the TSF by responding automatically such that the SFRs are always enforced.

FMT_MSA.2 Secure security attributes

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

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for:
security attributes of keys, PCR, NV storage areas and monotonic counter.

Note: The TOE supports the mechanism for updating the protected capabilities once the TOE is in the field as defined in the TPM_FieldUpgrade command of the TPM Main Specification. Within the scope of the TPM_FieldUpgrade command the security attributes of the TOE are also updated.

7.1.1 Extended Component FCS_RNG.1

To define the IT security functional requirements of the TOE an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes.

FCS_RNG.1 Random Number Generation

Hierarchical to: No other components

Dependencies: No dependencies

FCS_RNG.1 Random numbers generation Class DRG.3 according to [11]

FCS_RNG.1.1 The TSF shall provide a *deterministic* random number generator that implements:

(DRG.3.1) If initialized with a random seed using a PTRNG of class PTG.2 as random source, the internal state of the RNG shall have at least 100 bit of min-entropy.

(DRG.3.2) The RNG provides forward secrecy.

(DRG.3.3) The RNG provides backward secrecy even if the current internal state is known.

FCS_RNG.1.2 The TSF shall provide random numbers that meet:

(DRG.3.4) The RNG, initialized with a random seed during every startup and after 100 000 requests generates output for which more than 2^{34} strings of bit length 128 are mutually different with probability $w>1-2^{-16}$.

(DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A.

Application Note 2*:* The functional requirement FCS_RNG.1 is a refinement of the FCS_RNG.1 defined in the Protection Profile [8] according to “Anwendungshinweise und Interpretationen zum Schema (AIS)” [11].

Objects, Subjects, User Roles and Operations

The Subjects, User Roles Objects, and Operations used in the definition of the Security Functional Requirements are all defined in the PP [8] section 1.3.4.

7.2 Security Assurance Requirements

The security assurance requirements (SAR) of the TOE are the assurance components of the Evaluation Assurance Level 4 (EAL4) as defined in the Common Criteria [1] [2] [3] and augmented with *ALC_FLR.1* and *ADV_VAN.4*. They are all drawn from the Common Criteria V3.1 part 3. The security assurance components are listed in Table 3.

The security assurance requirements defined in Table 4 are defined in section 6.2 of the PP [8].

Table 3: Assurance components

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

7.3 Security Requirements Rationale

The security requirements rationale of the TOE are defined and described in the PP [8] section 6.3.

The security objectives O.Crypto_Key_Man and O.Reporting are covered by the security functional requirement FCS_CKM.1/AES.
O.Crypto_Key_Man is mapped additionally to:

- FCS_CKM.1/AES: Cryptographic key generation requires the TOE to generate AES keys with a key size of 128 bits according standards.

O.Reporting is mapped to:

- FCS_CKM.1/AES: Cryptographic key generation requires the TOE to generate AES keys with a key size of 128 bits.

The security functional requirement FCS_CKM.1/AES has the following dependencies:

- FCS_CKM.4 Cryptographic key destruction
  Rational: fulfilled
- FCS_CKM.2 Cryptographic key distribution, or
  FCS_COP.1 Cryptographic operation
  Rational: fulfilled by FCS_COP.1/SymEnc2

The security objective O.Crypto_Key_Man is covered by the security functional requirement FCS_COP.1/SymEnc2.
O.Crypto_Key_Man is mapped additionally to:

- FCS_COP.1/SymEnc2: Symmetric encryption and decryption in accordance with a specified cryptographic algorithm AES, modes CBC and CTR, and cryptographic key sizes of 128 bits.

The security functional requirement FCS_COP.1/SymEnc2 has the following 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
  Rational: fulfilled by FCS_CKM.1/AES
- FCS_CKM.4 Cryptographic key destruction
  Rational: fulfilled

8 TOE Summary Specification

The product overview is given in section 2.1. In the following the security functionality and the assurance measures of the TOE are described.

8.1 TOE Security Features

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

SF_CRY:	Cryptographic Support
SF_I&A:	Authentication and Identification
SF_ACC:	Access Control
SF_GEN	General
SF_P&T	Protection and Test

8.1.1 SF_CRY - Cryptographic Support

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

The TOE supports the generation of cryptographic keys in accordance with the specified cryptographic key generation algorithm *RSA key generator* and specified cryptographic key sizes RSA 512, 1024 and 2048 bits that meet the following: P1363 [P1363]. The source of randomness is the internal random generator (RNG).
The covered security functional requirement is FCS_CKM.1.

The TOE supports the generation of symmetric cryptographic keys in accordance with the specified cryptographic key generation algorithm *AES key generator* and specified cryptographic key sizes 128 bits.
The covered security functional requirement is FCS_CKM.1/AES.

The TOE supports the destruction of cryptographic keys by erasure of volatile memory areas containing cryptographic keys in accordance with FIPS PUB 140-1 [FIPS_140], section 4.8.5.
The covered security functional requirement is FCS_CKM.4.

The TOE performs the hash calculation in accordance with the specified cryptographic algorithm SHA-1 (cryptographic key sizes not available) that meets FIPS PUB 180-2 [FIPS_180].
The covered security functional requirement is FCS_COP.1/SHA.

The TOE performs HMAC calculation and verification in accordance with the specified cryptographic algorithm HMAC (SHA-1) and cryptographic key sizes 160 bits that meet RFC2104 [RFC2104] and FIPS PUB 180-2 [FIPS_180].
The covered security functional requirement is FCS_COP.1/HMAC.

The TOE performs signature generation and signature verification in accordance with the specified cryptographic algorithm RSA signature scheme [5] section 31.2.1, 31.2.2 and 31.2.3, and cryptographic key sizes RSA 512, 1024 and 2048 bits that meet PKCS#1 V2.0 [PKCS]. The TOE

uses the internal RNG as the source for any randomness that the process may require.
The covered security functional requirement is FCS_COP.1/RSA_Sig.

The TOE performs encryption and decryption in accordance with the specified cryptographic algorithm RSA encryption scheme [5] section 31.1.1, 31.1.2 and 31.1.3 and cryptographic key sizes RSA 512, 1024 and 2048 bits that meet PKCS#1v2.0 [PKCS].
The covered security functional requirement is FCS_COP.1/RSA_Enc.

The TOE performs the encryption and decryption in accordance with the specified cryptographic algorithm AES in the CBC mode and CTR mode and cryptographic key size of 128 bits that meet FIPS PUB 197 [FIPS_197], and the specified cryptographic algorithm MGF1 and cryptographic key sizes of 160 bits that meet PKCS#1v2.0 [PKCS].
The covered security functional requirement is FCS_COP.1/SymEnc and FCS_COP.1/SymEnc2.

The TOE provides a deterministic random number generator (DRNG) implemented in software to provide random numbers and a true random generator, which is used for the seeding of the DRNG. The TOE provides random numbers that fulfils the requirements from the functional class DRG.3 of [11].
The covered security functional requirement is FCS_RNG.1.

The SF_CRY “Cryptographic Support” covers the following security functional requirements: FCS_CKM.1, FCS_CKM.4, FCS_COP.1 and FCS_RNG.1.

8.1.2 SF_I&A - Authentication and Identification

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

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

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

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

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

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

The covered security functional requirements are FIA_UID.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.5, FIA_UAU.6 and FIA_AFL.1.

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

- FMT_MTD.1/AuthData
- FMT_MTD.1/Deleg
- FMT_MTD.1/Lock
- FMT_MTD.1/MigK

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

The covered security functional requirement is FIA_USB.1.

The SF_I&A “Authentication and Identification” covers the following security functional requirements: FIA_UID.1, FIA_UAU.1 FIA_UAU.4, FIA_UAU.5, FIA_UAU.6, FIA_AFL.1, FMT_MTD.1 and FIA_USB.1.

8.1.3 SF_ACC – Access Control

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

FDP_ACC.1/Modes
FDP_ACC.1/Deleg
FDP_ACC.1/KeyMan
FDP_ACC.1/MigK
FDP_ACC.1/M&R

FDP_ACC.1/NVS
FDP_ACC.1/MC
FDP_ACC.1/EID
FDP_ACC.1/DAA.

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

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

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

The covered security functional requirements are:

- FMT_MSA.1/Modes
- FMT_MSA.1/PhysP
- FMT_MSA.1/DFT
- FMT_MSA.1/DT
- FMT_MSA.1/KeyMan
- FMT_MSA.1/KEvi
- FMT_MSA.1/MigK
- FMT_MSA.1/MC
- FMT_MSA.1/DAA.

The TOE ensures that only secure values are accepted for security attributes.
The covered security functional requirement is FMT_MSA.2.

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

The covered security functional requirements are:

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

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

The covered security functional requirement is FDP_RIP.1.

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

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

The SF_ACC “Access Control” covers the following security functional requirements: FDP_ACC.1, FDP_ACF.1, FDP_ETC.2, FDP_ITC.2, RDP_RIP.1, FMT_MSA.1, FMT_MSA.2 and FMT_MSA.3.

8.1.4 SF_GEN – General

The TOE provides the roles: TPM owner, Entity owner, Delegated entity, Entity user, User using operatorAuth and “World” and associates users with roles. The role is bound always on specific authentication token, e.g. for the TPM owner it is the TPM ownership token and for the entity owner it is the entity token. The roles are enforced within the TOE because there are specific commands and specific keys bond to different token.
The covered security functional requirement is FMT_SMR.1.

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

The covered security functional requirement is FMT_SMF.1.

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

The covered security functional requirement is FPT_TDC.1.

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

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

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

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

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

The covered security functional requirement is FAU_GEN.1.

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

The covered security functional requirement is FPT_STM.1.

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

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

The SF_GEN “General” covers the following security functional requirements: FMT_SMR.1, FMT_SMF.1, FDP_TDC.1, FDP_UCT.1, FDP_UIT.1, FPT_STM.1, FCO_NRO.1 and FAU_GEN.1.

8.1.5 SF_P&T – Protection and Test

The TOE preserves a secure state when a failure of any crypto operations including RSA encryption, RSA decryption, AES encryption, AES decryption, SHA-1, RNG, RSA signature generation, HMAC generation or failure of any commands or internal operations and authorization occurs.

The covered security functional requirement is FPT_FLS.1.

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

The covered security functional requirement is FPT_TST.1.

The TOE supports the Direct Anonymous Attestation Protocol.

The covered security functional requirement is FPR_UNL.1.

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

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

- The correct function of the TOE is only given in the specific range of the environmental operating parameters. To prevent an attack exploiting those circumstances the external clock conditions, the temperature and electro magnetic radiation (e.g. light) are observed to detect if the specified range is left. The TOE falls into the defined secure state in case of a specified range violation[2]. The defined secure state causes the chip internal reset process.
- The data in the EEPROM are automatically monitored by the EDC. In case of a 1 bit error the memory content is corrected by the ECC, in case of more bit errors the TOE enters the secure state.
- Several mechanisms protect the TOE against snooping the design or the user data during operation and even if it is out of operation (power down). There are topological design measures for disguise, such as the protection of security critical lines by specific intelligent and intrinsic shielding including secure wiring of security critical signals. The entire design is kept in a non standard way to prevent attacks using standard analysis methods. A dedicated CPU with a non public bus protocol is used which makes analysis complicated.
- The readout of data can be controlled with the use of encryption. An attacker can not use the data obtained by espionage due to their encryption. The memory contents of the TOE are encrypted on chip to protect against data analysis on stored data as well as on internally transmitted data.
- The life cycle of the TOE is split-up in several phases. The development (phase 1), the manufacturing (phase 2), the TOE preparation (phase 3, 4 and 5) and the TOE operational (phase 6 and 7) is a rough split-up from TOE point of view. These phases are implemented in the TOE as Test Mode (phase 1 and 2) and User Mode (phase 2 to 6). During start-up of the TOE the decision for the User Mode or the Test Mode is taken dependent on several phase identifiers. If Test Mode is the active phase the TOE requests authentication before any action.
- The virtual physical address mapping together with the memory management unit **(**MMU) gives the operating system the possibility to define different access rights for memory areas. In case of an access violation the MMU will generate a non maskerable interrupt (NMI) and an interrupt service routine react on the access violation.
- The covered security functional requirement is FPT_PHP.3.

The SF_P&T “Protection & Test” covers the following security functional requirements:
FPT_FLS.1, FPR_UNL.1, FPT_PHP.3 and FPT_TST.1

[2] The operating state checking this functionality can only work when the TOE is running and can not prevent reverse engeniering.

8.1.6 Assignment of Security Functional Requirements

The justification of the mapping between security functional requirements and the security features is given in sections 8.1.1 – 8.1.5. The results are shown at Table 4.

Security Functional Requirement	SF_ CRY	SF_ I&A	SF_ ACC	SF_ GEN	SF_ P&T
FMT_SMR.1				X	
FMT_SMF.1				X	
FDP_ACC.1/Modes			X		
FDP_ACC.1/Deleg			X		
FDP_ACC.1/KeyMan			X		
FDP_ACC.1/MigK			X		
FDP_ACC.1/M&R			X		
FDP_ACC.1/NVS			X		
FDP_ACC.1/EID			X		
FDP_ACC.1/MC			X		
FDP_ACC.1/DAA			X		
FDP_ACF.1/Modes			X		
FDP_ACF.1/Deleg			X		
FDP_ACF.1/KeyMan			X		
FDP_ACF.1/MigK			X		
FDP_ACF.1/M&R			X		
FDP_ACF.1/NVS			X		
FDP_ACF.1/MC			X		
FDP_ACF.1/EID			X		
FDP_ACF.1/DAA			X		
FMT_MSA.1/Modes			X		
FMT_MSA.1/PhysP			X		
FMT_MSA.1/DFT			X		
FMT_MSA.1/DT			X		
FMT_MSA.1/KeyMan			X		
FMT_MSA.1/MigK			X		
FMT_MSA.1/Kevi			X		
FMT_MSA.1/MC			X		

FMT_MSA.1/DAA			X		
FMT_MSA.2			X		
FMT_MSA.3/Deleg			X		
FMT_MSA.3/KeyMan			X		
FMT_MSA.3/M&R			X		
FMT_MSA.3/NVS			X		
FMT_MSA.3/MC			X		
FMT_MSA.3/EID			X		
FMT_MSA.3/DAA			X		
FDP_ETC.2			X		
FDP_ITC.2			X		
FDP_RIP.1			X		
FCS_CKM.1	X				
FCS_CKM.1/AES	X				
FCS_CKM.4	X				
FCS_RNG.1	X				
FCS_COP.1/SHA	X				
FCS_COP.1/HMAC	X				
FCS_COP.1/RSA_Sig	X				
FCS_COP.1/RSA_Enc	X				
FCS_COP.1/SymEnc	X				
FCS_COP.1/SymEnc2	X				
FMT_MTD.1/AuthData		X			
FMT_MTD.1/Deleg		X			
FMT_MTD.1/Lock		X			
FMT_MTD.1/MigK		X			
FIA_UID.1		X			
FIA_UAU.1		X			
FIA_UAU.4		X			
FIA_UAU.5		X			
FIA_UAU.6		X			
FIA_AFL.1		X			
FIA_USB.1		X			
FPT_TDC.1				X	
FCO_NRO.1/M&R				X	
FCO_NRO.1/STS				X	

FDP_UCT.1/Exp				X	
FDP_UCT.1/Imp				X	
FDP_UIT.1/Data				X	
FDP_UIT.1/Sesson				X	
FAU_GEN.1				X	
FPT_STM.1				X	
FPT_FLS.1					X
FPT_TST.1					X
FPR_UNL.1					X
FPT_PHP.3					X

Table 4: Assignment security functional requirement to security features

8.2 Security Function Policy

The TOE enforces user access to cryptographic IT assets in accordance with the following security function policies (SFP)

- "TPM Mode Control SFP" (MCT-SFP)
- "Delegation" (Del-SFP)
- "Key Management SFP" (KeyM-SFP)
- "Key Migration SFP" (KMig-SFP)
- "Measurement and Reporting SFP" (M&R-SFP)
- "Non-volatile Storage SFP" (NVS-SFP)
- "Monotonic Counter SFP" (MC-SFP)
- "Export and Import of Data SFP" (EID-SFP)
- "Direct Anonymous Attestation Protocol SFP (DAA-SFP)

to meet the security functional requirements.

These policies include different operational roles, subjects, objects and operations which are described in the following.

8.2.1 Operational roles

The following objects are defined:

(1) TPM owner

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

(2) Delegated entity

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

(3) Entity owner

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

(4) Entity user

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

(5) User using operatorAuth

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

(6) World

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

The TPM knows some user roles by permanently stored individual authentication data: the TPM owner using ownerAuth and a user role defined by operatorAuth[3]. Other user roles are object specific (entity owner and entity user). The users may use these roles except “World” after authentication only. The user establishing an OSAP session negotiates the authorization handle and the shared secret as user authentication data during this session.

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

- Physical presence
  The TCG policies require physical presence to be indicated to the TPM to enable the user to execute privileged commands or to alter the following states if no TPM owner is defined: disabled, deactivated, and (ownership-)disabled (cf. [5], section 10, and section 6.1.3 TPM Operational Mode of the PP [8] for further details). The TPM and platform manufacturer defines the way how the TPM physical presence is asserted by hardware input. The physical presence may be asserted by means of the command TSC_PhysicalPresence (cf. [7], section 6.6).

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

8.2.2 Subjects

The following subjects are defined:

(1) The initialization process initiated by the TPM_Init signal of the platform to the TPM.

(2) The self test which is started after initialization and continued after the TPM_ContinueSelfTest command is received.

(3) A process receiving, executing and responding to a single command.

(4) A session (sequence of commands) which may establish shared secrets, encryption keys, and session logs. The session may be an authorization session (cf.[5], section 13), monotonic counter sessions (cf. [5], section 20.1), a tick session (cf. [5], section 20.1) or a transport session (cf. [5], section 18).

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

[3] This user is some times reffered to as „operator“

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

The subject is associated with the security attributes

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

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

8.2.3 Objects, Operations and Security Attributes

A detailed description is given in PP [8] section 1.3.4 and Table 1.

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

The Table 1 of the PP [8] lists the objects, the operation via reference to the commands as described in the TPM specification [7] and the security attributes of the objects as described in the TPM specification [6].

The objects listed Table 1 of the PP [8] are user data except the EK and SRK.

The TSF data are defined as:

- EK and SRK,
- authentication reference data of the TPM owner, the entity using operatorAuth, delegated entities, owner of entities, user of entities, and
- TPM flags disable, deactivated and ownership as part of the TPM_PERMANENT_FLAGS,

security attributes of the objects and subjects.

9 Reference

9.1 Literature

[FIPS_197] FIPS PUB 197, Advanced Encryption Standard (AES), November 26, 2001 U.S. Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory (ITL)

[FIPS_140] FIPS PUB 140-1, Security Requirements for Cryptographic Modules, January 11, 1994, U.S. Department of Commerce, National Institute of Standards and Technology

[FIPS_180] FIPS PUB 180-2, Secure Hash Standard, August 1, 2002, U.S. Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory (ITL)

[RFC2104] RFC 2104: HMAC: Keyed-Hashing for Message Authentication, http://www.ietf.org/rfc/rfc2104.txt

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

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

[N800] NIST Special Publication 800-38A, 2001 Edition, Recommendation for Block Cipher Modes of Operation Methods and Techniques

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 4, CCMB-2012-09-001, September 2012

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 3.1, Revision 4, CCMB-2012-09-002, September 2012

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 4, CCMB-2012-09-003, September 2012

[4] Common Methodology for Information Technology Security Evaluation Methodology, Evaluation Methodology, Version 3.1, Revision 4, CCMB-2012-09-004, September 2012

[5] TPM Main Part 1 Design Principles, Specification Version 1.2, Revision 116, 1 March 2011, Trusted Computing Group, Incorporated

[6] TPM Main Part 2 TPM Structures, Specification Version 1.2, Revision 116, 1 March 2011, Trusted Computing Group, Incorporated

[7] TPM Main Part 3 Commands, Specification Version 1.2, Revision 116, 1 March 2011, Trusted Computing Group, Incorporated

[8] Trusted Computing Group Protection Profile PC Client Specific Trusted Platform Module TPM Family 1.2; Level 2 Revision 116, Version 1.2; 18 May, 2011

[9] TCG PC Client Specific TPM Interface Specification (TIS) for TPM Family 1.2; Level 2, Version 1.21 FINAL, Revision 1.00, May 2011

[10] OPTIGA™ TPM SLB 9660 TPM1.2 Databook, Infineon Technologies AG, Revision 1.4, 2017-06-26

[11] Anwendungshinweise und Interpretationen zum Schema (AIS), AIS 20 Version 2.1, Bundesamt für Sicherheit in der Informationstechnik

[63] TPM Trusted Platform Module Version 1.2 SLB9660 Basic Platform Manufacturer Guideline, Infineon Technologies AG, V1.00, Edition 2013-03

[88] OPTIGA™ TPM SLB 9660 TPM1.2 Errata and Updates, Infineon Technologies AG, Revision 1.9, 2017-06-26

9.2 List of Abbreviations

CC - Common Criteria
CI - Chip Identification mode (STS-CI)
CIM - Chip Identification Mode (STS-CI), same as CI
CRC - Cyclic Redundancy Check
DPA - Differential Power Analysis
DFA - Differential Failure Analysis
DRNG - Deterministic Random Number Generator
EAL - Evaluation Assurance Level
ECC - Error Correction Code
EDC - Error Detection Code
EEPROM - Electrically Erasable and Programmable Read Only Memory
EMA - Electro magnetic analysis
HW - Hardware
IC - Integrated Circuit
ID - Identification
IRAM - Internal Random Access Memory
IT - Information Technology
I/O - Input/Output
MED - Memory Encryption and Decryption
MMU - Memory Management Unit
OS - Operating system
PLL - Phase Locked Loop
PP - Protection Profile
RMS - Resource Management System
RNG - Random Number Generator
RAM - Random Access Memory
ROM - Read Only Memory
SF - Security Feature
SFP - Security Function Policy
SFR - Special Function Register
SPA - Simple power analysis
ST - Security Target
STS - Self Test Software
SW - Software
TM - Test Mode (STS)
TOE - Target of Evaluation
TSF - TOE Security Functionality
TSP - TOE Security Policy
UM - User Mode (STS)
XRAM - eXtended Random Access Memory

9.3 Glossery

Blob: Opaque data of fixed or variable size. The meaning and interpretation of the data is outside the scope and context of the Subsystem.

Central Processing Unit(CPU): Logic circuitry for digital information processing.

Chip → Integrated Circuit

Chip Identification Mode: Operational status phase of the TOE, in which actions for identifying the individual take place.

Controller: IC with integrated memory, CPU and peripheral devices.

CRC: Process for calculating checksums for error detection.

Challenger: An entity that requests and has the ability to interpret integrity metrics from a Subsystem.

EEPROM: Nonvolatile memory permitting electrical read and write operations.

Endorsement Key: A term used ambiguously, depending on context, to mean a pair of keys, or the public key of that pair, or the private key of that pair; an asymmetric key pair generated by or inserted in a TPM that is used as proof that a TPM is a genuine TPM; the public endorsement key (PUBEK); the private endorsement key (PRIVEK).

Firmware: Part of the software implemented as hardware.

Hardware: Physically present part of a functional system.

Hash value: Result of a hash calculation e.g. SHA-1.

HMAC: A mechanism for message authentication according RFC 2104 using the cryptographic hash function SHA-1.

Integrity metrics: Values that are the results of measurements on the identity for the TPM.

Integrated Circuit: Component comprising several electronic circuits implemented in a highly miniaturized device using semiconductor technology.

Internal Random Access Memory: RAM integrated in the CPU.

LPC Interface: Low Pin Count (LPC) Interface defined by Intel is a standardized interface used in PC mainboards.

Man-in-the-middle attack: An attack by an entity intercepting communications between two others without their knowledge and by intercepting that communication able to obtain or modify the information between them.

Mechanism: Logic or algorithm which implements a specific security function in Hardware or software.

Memory: Hardware part containing digital information (binary data).

Memory Encryption and Decryption: Method of encoding/decoding data transfer between CPU and memory.

Memory Management Unit (MMU): The MMU controls the different access rights of memory areas.

Microcontroller → Controller

Microprocessor → CPU

Migratable: A key that may be transported outside the specific TPM.

Nonce: A nonce is a random number value that provides protection from replay and other attacks.

Non-migratable: A key that cannot be transported outside the specific TPM. A key that is (statistically) unique to a particular TPM.

Owner: The entity that owns the platform in which a TPM is installed. Since there is, by definition, a one-to-one relationship between the TPM and the platform, the Owner is also the Owner of the TPM. The Owner of the platform is not necessarily the "user" of the platform (e.g., in a corporation, the Owner of the platform might be the IT department while the user is an employee). The Owner has administration rights over the TPM.

Platform Configuration Register (PCR): A PCR consists of a 160 bit field that holds a cumulatively updated hash value and a 4 byte status field.

Private Endorsement Key (PRIVEK): The private key of the key pair that proves that a TPM is a genuine TPM. The PRIVEK is (statistically) unique to only one TPM.

Protected function: Access to this function requires an authentication process.

Public Endorsement Key(PUBEK): The public key that proves that a TPM is a genuine TPM. The PUBEK is (statistically) unique to only one TPM.

Protection Profile: A document that defines all attacks and how they are resisted by the TPM, the RTM, and the methods by which these are incorporated into the platform.

Random Access Memory: Volatile memory which permits write and read operations.

Random Number Generator: Hardware part for generating random numbers.

Read Only Memory: Nonvolatile memory which permits read operations only.

Resource Management System: Part of the firmware containing EEPROM programming routines.

Root of Trust for Measurement(RTM): The point from which all trust in the measurement process is predicated.

Root of Trust for Reporting(RTR): The point from which all trust in reporting of measured information is predicated.

Root of Trust for Storing(RTS): The point from which all trust in Protected Storage is predicated.

RSA: An asymmetric encryption method using two keys: a private key and a public key. Reference: http://www.rsa.com.

SAM: Service Algorithm Minimal

Security Feature: Part(s) of the TOE used to implement part(s) of the security objectives.

Security Target: Description of the intended state for countering threats.

Self Test Software: Part of the firmware with routines for controlling the operating state and testing the TOE hardware.

SHA-1: A hashing algorithm producing a 160-bit result from an arbitrary source as specified in FIPS 180-1.

Shielded location: Storage location within the TPM with a protection against unauthorized access.

Smart Card: Plastic card in credit card format with built-in chip.

Storage Root Key (SRK): The root key of a hierarchy of keys associated with a TPM; generated within a TPM; a non-migratable key.

Subsystem: The combination of the TSS and the TPM.

Software: Information (non-physical part of the system) which is required to implement functionality in conjunction with the hardware (program).

Target of Evaluation: Product or system which is being subjected to an evaluation.

Test Mode: Operational status phase of the TOE in which actions to test the TOE hardware take place.

Threat: Action or event that might prejudice security.

TpmProof: A random number stored within the TPM. The tpmProof is a unique secret for each TPM.

Trusted Platform Module: The set of functions and data that are common to all types of platform, which must be trustworthy if the Subsystem is to be trustworthy; a logical definition in terms of protected capabilities and shielded locations.

Trusted Platform Support Services (TSS): The set of functions and data which are common to all types of platform, which are not required to be trustworthy (and therefore do not need to be part of the TPM).

TCG-protected capability: A function that is protected within the TPM, and has access to TPM secrets.

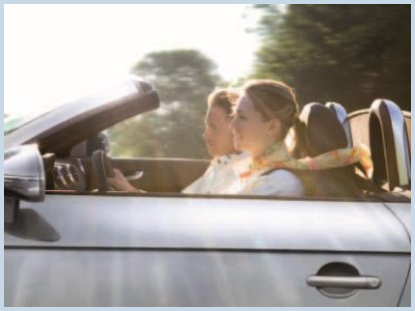
Trusted Set (TS): Subsystem capability that must be trustworthy for the subsystem.

TPM Identity: One of the anonymous PKI identities belonging to a TPM; a TPM may have multiple identities.

User: An entity that uses the platform in which a TPM is installed. The only rights that a User has over a TPM are rights given to the User by the Owner. These rights are expressed in the form of authentication data, given by the Owner to the User, that permits access to entities protected by the Owner of the platform (e.g. in a corporation, the owner of the platform might be the IT department while the User is an employee). There can be multiple Users.

User Mode: Operational status phase of the TOE in which actions intended for the user take place.

Infineon Technologies – innovative semiconductor solutions for energy efficiency, mobility and security.



www.infineon.com

Published by Infineon Technologies AG