Infineon Technologies AG CCS ESS

Trusted Platform Module 2.0
SLB 9670

FIPS 140-2 Level 2 Non-Proprietary Security Policy

Version: 1.00
Date: 07 February 2018

Table of Contents

List of Tables

List of Figures

References

Table 1: References

Acronym	Full Specification Name
[FIPS 180-4]	NIST, *Secure Hash Standard*, FIPS Publication 180-4, March 2012
[FIPS 186-4]	NIST, *Digital Signature Standard (DSS)*, FIPS Publication 186-4, July 2013
[FIPS140-2]	NIST, *Security Requirements for Cryptographic Modules*, May 25, 2001
[FIPS197]	NIST, *Advanced Encryption Standard (AES)*, FIPS Publication 197, November 26, 2001
[FIPS 198-1]	NIST, *The Keyed-Hash Message Authentication Code (HMAC),* July 2008
[IG]	NIST, *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*
[PKCS#1]	*PKCS #1 v2.1: RSA Cryptography Standard*, RSA Laboratories, June 14, 2002
[SP800-38A]	NIST Special Publication SP 800-38A, *Recommendation for Block Cipher Modes of Operation,* 2001
[SP800-56A]	NIST Special Publication SP 800-56A, Revision 1, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography,* March 2007
[SP800-56B]	NIST Special Publication SP 800-56B, Revision 1, *Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography,* September 2014
[SP 800-90A]	NIST Special Publication 800-90A Revision 1, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, June 2015
[SP800-108]	NIST, *Recommendation for Key Derivation Using Pseudorandom Functions (Revised),* October 2009
[SP800-131A]	*Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, January 2011
[TPM_LIB]	TCG, *TPM Library Part 1 Architecture, Family "2.0", Level 00*, Revision 01.16, 30 October 2014 TCG, *TPM Library Part 2 Structures, Family "2.0", Level 00,* Revision 01.16, 30 October 2014 TCG, *TPM Library Part 3 Commands, Family "2.0", Level 00*, Revision 01.16, 30 October 2014 TCG, *TPM Library Part 4 Supporting Routines, Family "2.0", Level 00*, Revision 01.16, 30 October 2014
[TPM_PTP]	TCG, PC Client Platform TPM Profile (PTP) Specification, Family "2.0", Revision 00.43, January 26, 2015

Acronyms and Definitions

Table 2: Acronyms and Definitions

Acronym	Definition
CCS ESS	The Infineon group: Chip Card and Security, Embedded Security Solutions
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRNGT	FIPS 140-2 AS09.42 Continuous Random Number Generator Test
EEPROM	Electrically Erasable Programmable Read-Only Memory
IC	Integrated Circuit
KAT	Known Answer Test
KAS	Key Agreement Scheme
KBKDF	Key Based Key Derivation Function
MED	Memory Encrypt/Decrypt Unit
MMU	Memory Management Unit
NVM	Non-Volatile Memory (e.g., EEPROM, Flash)
PCT	Pairwise Consistency Test
RAM	Random-Access Memory
ROM	Read-Only Memory
SPI	Serial Peripheral Interface; Motorola / de-facto standard for a synchronous serial communication interface. An alternative to the LPC for the TPM
TCG	Trusted Computing Group (http://www.trustedcomputinggroup.org/)
TPM	Trusted Platform Module
TRNG	True Random Number Generator (a form of hardware random number generator)

1 Overview

This document defines the Security Policy for the Infineon Trusted Platform Module 2.0 SLB 9670 cryptographic module, hereafter denoted *TPM*. The TPM, validated to FIPS 140-2 overall Level 2, is a single chip module 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 TPM is a complete solution implementing the TCG specifications for the TPM 2.0 family, [TPM_LIB] and [TPM_PTP]. See http://www.trustedcomputinggroup.org/ for further information on TCG and TPM.

The TPM is designated as a non-modifiable operational environment under the FIPS 140-2 definitions. The FIPS 140-2 security levels for the TPM are as follows:

Table 3: Security Level of Security Requirements

Security Requirement	Level
Cryptographic Module Specification	2
Cryptographic Module Ports and Interfaces	2
Roles, Services, and Authentication	2
Finite State Model	2
Physical Security	3
Operational Environment	N/A
Cryptographic Key Management	2
EMI/EMC	3
Self-Tests	2
Design Assurance	2
Mitigation of Other Attacks	2

1.1 Versions, Configurations and Modes of Operation

Table 4: Configuration Part and Version Numbers

HW Part	Package	Firmware Version
SLB 9670	PG-UQFN-32-1	7.83
SLB 9670	PG-VQFN-32-13	7.83

The TPM is intended for use in general purpose computing environments, as a device peripheral to the CPU, with the application controlling the usage of the module. The TPM is operated in the FIPS 140-2 Approved mode when the application complies with the conditions listed in Section 8.1.

The TPM provides two Modes of Operations: Platform Initialization Mode and Full Operational Mode. In Platform Initialization Mode mainly Random Number Generation, Hash services and Password Verification Authentication Mechanism are available. In Full Operational Mode all services and authentication mechanisms are available. The Platform Initialization Mode is entered after the module is powered up or reset. The Full Operational Mode is entered either explicitly via the command TPM2_SelfTest or implicitly via the call of a service which uses Algorithms only available in Full Operational Mode. If the requirements for secure operation from section 8.1 are met, both Platform Initialization Mode and Full Operational Mode are Approved modes of operation in the meaning of FIPS 140-2. For the detailed differences please refer to section 2, section 3.2 and section 4.

The security functions possible in the non-Approved mode are listed in Table 9.

The *Show Status* service (specifically TPM2_GetCapability with the capability=TPM_CAP_TPM_PROPERTIES and property=TPM_PT_FIRMWARE_VERSION_1 qualifier) may be used to verify the FIPS-compliant version of TPM firmware is present in the TPM.

1.2 Physical Characteristics and Cryptographic Boundary

The TPM cryptographic boundary is the surfaces, edges and connection points of the IC package, PG-UQFN-32-1 (see Figure 1) or PG-VQFN-32-13 (see Figure 2). The packages are shown in the figures on a 1 mm by 1 mm grid to indicate size. The physical ports and logical interfaces are detailed in Table 6.

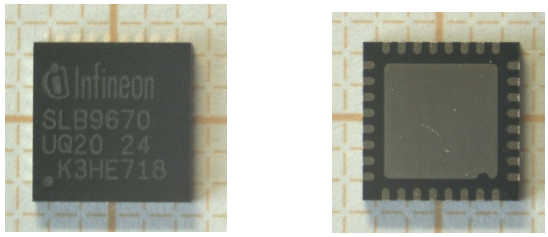

Figure 1: TPM 2.0 in package PG-UQFN-32-1 (left: top view; right: bottom view)

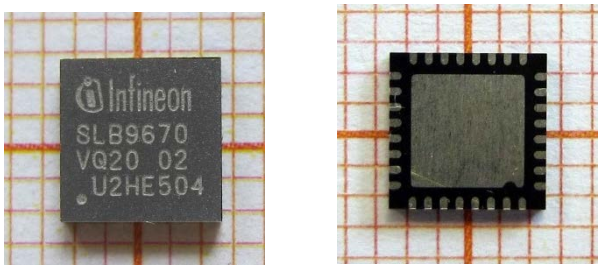

Figure 2: TPM 2.0 in package PG-VQFN-32-13 (left: top view; right: bottom view)

Table 5: Types printed on packages

Package	HW Part	Type printed on package
PG-UQFN-32-1	SLB 9670	SLB9670UQ20 (shown in Figure 1) SLB9670XU20
PG-VQFN-32-13	SLB 9670	SLB9670VQ20 (shown in Figure 2) SLB9670VQ12 SLB9670XQ20 SLB9670XQ12

Table 6: Ports and Interfaces

Port	Ports common to all configurations	Logical Interface Type
GND	Ground	Power
GPIO	General Purpose I/O	Control Input, Status Output
NC	No connects	Unused
PP	Physical Presence	Control Input
VDD	1.8V or 3.3V	Power
SPI Interface Specific (SLB 9670) Ports and mapping to Logical Interfaces		
MISO	Master Input, Slave Output	Control Input, Data Output, Status Output
MOSI	Master Output, Slave Input	Control Input, Data Input, Status Output
SCLK	Serial clock	Control Input
SS	Slave Select	Control Input

1.3 Operational Environment

The module has a non-modifiable operational environment under the FIPS 140-2 definitions. The module includes a firmware load function to support necessary updates. New firmware versions within the scope of this validation must be validated through the FIPS 140-2 CMVP. Any other firmware loaded into this module is out of the scope of this validation and requires a separate FIPS 140-2 validation.

1.4 TPM Composition

Figure 3 depicts the TPM hardware block diagram, shown from a logical perspective. The red outline indicates the cryptographic boundary.

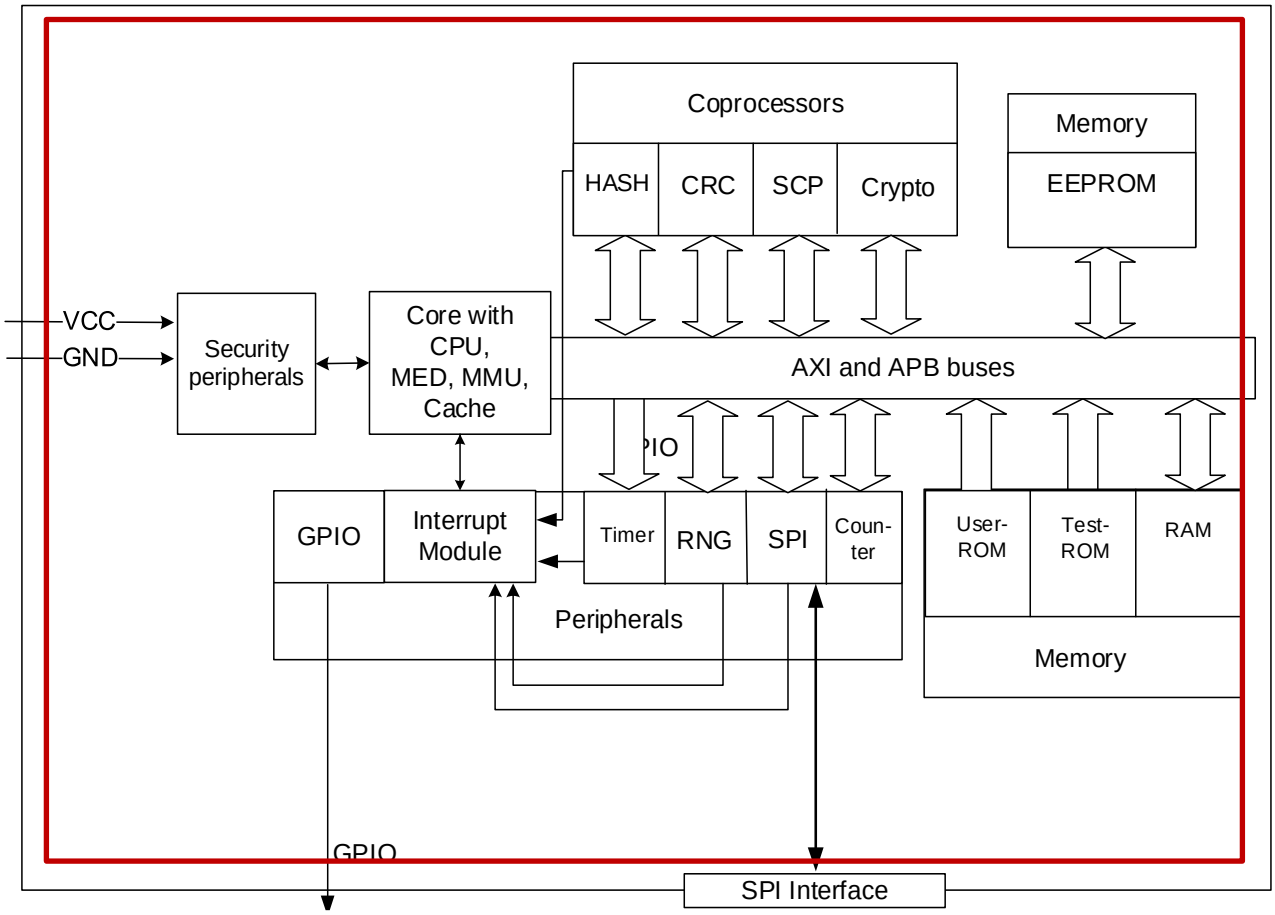

Figure 3: Module Block Diagram

The major blocks of the TPM are:

- Core: Dual CPU (configured to continuously detect faults and assure calculation integrity); MMU (memory management with privilege levels); MED (Memory Encrypt/Decrypt) and cache.
- Hardware accelerators (coprocessors): SCP (symmetric co-processor) for AES hardware acceleration; an Asymmetric Crypto Co-processor (labeled Crypto in Figure 3) for modular math (e.g. RSA 2048-bit, ECC) acceleration; a SHA-1/SHA-256 accelerator, labelled HASH in the figure; The checksum module (labeled CRC in Figure 3) allows simple calculation of 16-bit CRC checksums.
- Memory: ROM, EEPROM and RAM.
- Peripherals: timer; counter; a physical, non-deterministic random number generator called TRNG (True Random Number Generator); and the physical ports that cross the cryptographic boundary (SPI for SLB 9670).
- The SPI block corresponds to the interfaces in above.
- Security peripherals: Security logic, shield, an interrupt-controlled I/O interface.
- GPIO: Bidirectional signal which can be set or read by the caller; not otherwise used by the TPM.
- The processor firmware provides the TCG functionality specified in [TPM_LIB]; these are the set of services described in Section 3.2.

2 Cryptographic Functionality

The TPM implements the approved and allowed cryptographic functions listed in Table 7 and Table 8.
All Cryptographic Functions are available in Full Operational Mode, Cryptographic Functions available in Platform Initialization (PI) Mode are formatted bold and marked with an index "PI". For a detailed description of the Platform Initialization Mode please refer to section 1.1.

Table 7: Approved Cryptographic Functions

CAVP Cert	Algorithm	Standard(s)	Mode/Method	Key Lengths/ Curves/Moduli	Use
5069	AES	FIPS 197, SP800-38A	CFB	128 bit	Data Encryption/Decryption
Vendor affirmed	CKG	SP 800-133	Section 6.1 Asymmetric signature key generation using unmodified DRBG output Section 6.2 Asymmetric key establishment key generation using unmodified DRBG output Section 7.1 Direct symmetric key generation using unmodified DRBG output Section 7.3 Derivation of symmetric keys from a key agreement shared secret. Section 7.4 Derivation of symmetric keys from a pre-shared key		Key Generation
1629	CVL ECC-CDH	SP 800-56A	--	P-256	Key Agreement Primitive
1630	CVL RSADP	SP 800-56B	--	2048 bit	Key Transport Primitive
1886	**DRBG** [PI)]	SP 800-90A	CTR_DRBG: AES-128 (with derivation function)	security strength 128 bit	Deterministic Random Bit Generation
1314	ECDSA	FIPS 186-4	--	P-256	Key Generation, Public Key Validation
			SHA-256	P-256	Signature Generation, Signature Verification
			SHA-1	P-256	Signature Verification
3383	**HMAC** [PI)]	FIPS 198-1	HMAC-SHA-1, **HMAC-SHA-256** [PI)]	160 bit, 256 bit	Message Authentication
157	KAS EC-DH	SP 800-56A	One-Pass DH, Single-Step KDF (SHA-1, SHA-256), Initiator/Responder, No Key Confirmation	P-256	Key Generation, Partial Public Key Validation, Key Agreement
172	KBKDF	SP 800-108	CTR mode HMAC-SHA1, HMAC-SHA256		Key Derivation

CAVP Cert	Algorithm	Standard(s)	Mode/Method	Key Lengths/ Curves/Moduli	Use
5069 & 3383	KTS AES & HMAC	FIPS 197, SP 800-38F	CFB & HMAC-SHA1, HMAC-SHA256	128 bit & 160 bit, 256 bit	Key Wrapping/Unwrapping
Vendor affirmed	KTS RSA	SP 800-56B	KTS-OAEP-basic	2048 bit	Key Transport
2749	RSA	FIPS 186-4, PKCS#1 v2.1	--	2048 bit	Key Generation
			SHA-256 & PKCS1-V1_5, PSS	2048 bit	Signature Generation, Signature Verification
			SHA-1 & PKCS1-V1_5, PSS	2048 bit	Signature Verification
4129	**SHS** [PI]	FIPS 180-4	SHA-1, SHA-256		Message Digest

Table 8: Allowed Cryptographic Functions

Algorithm	Caveat	Use
NDRNG [PI]- entropy source internal to the module's cryptographic boundary	None	Seeding/Reseeding of the DRBG

The following table shows the cryptographic functions of TPM, which are neither Approved nor allowed.

Table 9: Non-Approved Cryptographic Functions

Algorithm	Use
ECDAA	For Elliptic Curve Direct Anonymous Attestation used to generate anonymous signatures with the Barreto-Naehrig (BN) elliptic curve BN-256
RSA 1024-bit (non-compliant)	Key Generation, Signature Generation, Signature Verification, Key Wrapping (non-compliant because less than 112 bits of encryption strength)
SHA-1 (non-compliant)	For subsequent Signature Generation

2.1 Critical Security Parameters and Public Keys

All CSPs used by the Module are listed in Table 10.

Table 10: Cryptographic Keys and CSPs

Name	Description and usage
DRBG-EI	TPM DRBG Entropy Input - produced by the NDRNG, used during DRBG instantiation and reseed.
DRBG-STATE	TPM DRBG State - Current values of AES 128 CTR_DRBG state (V and K).
TPM-EPS	TPM Endorsement Primary Seed - Minimum of 256 bit random value used as a master seed value to derive primary keys and secrets in the Endorsement Hierarchy; installed at the factory.
TPM-PS	TPM Platform, Storage Primary Seed – Minimum of 256 bit random value; used as a master seed value to derive hierarchy primary keys and secrets in the Platform respective Storage Hierarchy.
TPM-Proof	TPM Proof Value - 256 bit random value used as KW-KDK Key in an HMAC-SHA256 KBKDF (to derive confidentiality keys KW-CK) and used as KW-IK for HMAC-SHA256 Integrity Protection of CSP wrapping in the TPM Context Management Service. Also used as HMAC-SHA256 Integrity Key to prove that data structures, have only been generated by a specific TPM module.
AS-AD-K	Authorization Session Authentication Data Key - 160 bit or 256 bit secret authentication data known by the Object Owner or Hierarchy Owner. E.g. used to derive TPM-AS-SK for Bound Authorization Sessions.
AS-SALT	Authorization Session Salt - 160 bit or 256 bit Salt value used to derive a TPM-AS-SK for Salted Authorization Sessions.
AS-SK	Authorization Session - Session Key - HMAC-SHA1 160 bit or HMAC-SHA256 bit session key used for message authentication in a bound and/or salted TPM Authorization Session.
ES-KDK	Encryption Session Key Derivation Key - 160 bit or 256 bit Key used to derive ES-EK.
ES-EK	Encryption Session Ephemeral Key - AES 128 Bit ephemeral key used for message parameter encrypt/decrypt within a secure messaging session.
KW-KDK	Key Wrap - Key Derivation Key - 160 bit or 256 bit secret used to derive KW-IK and KW-CK.
KW-IK	Key Wrap - Integrity Key - HMAC-SHA1 160 bit or HMAC-SHA256 Key used for integrity protection of encrypted data used in the TPM for Key Wrapping Mechanism:
KW-CK	Key Wrapping - Confidentiality Key - AES 128 bit key used to protect CSP confidentiality via Key Wrapping used in the TPM.
SIGK	Signing Key - RSA 2048 bit or ECC P-256 private key used for TPM Protocols and User Signature Generation Services.
IFX-PE-KEK	Infineon Primary Endorsement Key Establishment Key - RSA 2048 bit (key transport) or ECC P-256 (key agreement) private key used in TPM Protocols and uniquely associated with each TPM device via an Infineon X509 Certificate; installed at the factory.
TPM-KEK	TPM Key Establishment Key - RSA 2048 bit (key transport) or ECC P-256 (key agreement) private key used in TPM Protocols.
U-KEK	User Key Establishment Key - RSA 2048 bit (key transport) or ECC P-256 (key agreement) private key used in a User Key Agreement Primitive Services.
U-MACK	User HMAC Key - 160 bit or 256 bit HMAC key used in User HMAC Services.
U-E-KAK	User Ephemeral Key Agreement Key - ECC P-256 bit private key used in a User Key Agreement Primitive

Name	Description and usage
	Services

Table 11: Public Keys

Name	Description and usage
SIGK-PUB	Public Signing Key - RSA 2048 bit or ECC P-256 public signature verification key used for TPM Enhanced Authorization Protocols and User Signature Verification Service.
TPM-KEK-PUB	TPM Public Key Establishment Key - RSA 2048 bit (key transport) or ECC P-256 (key agreement) key used in TPM Protocols.
U-KEK-PUB	User Public Key Establishment Key - RSA 2048 bit (key transport) or ECC P-256 (key agreement) key used in a User Key Agreement Primitive Services.
U-E-KAK-PUB	User Public Ephemeral Key Agreement Key - ECC P-256 bit ephemeral public key used in a User Key Agreement Primitive Services.
IFX-SIGK-PUB	Infineon Public Signing Key - RSA 2048 bit public key for field upgrade (firmware load test) signature verification; installed at the factory.
IFX-PE-KEK-PUB	Infineon Public Primary Endorsement Key Establishment Key - RSA 2048 bit (key transport) or ECC P-256 (key agreement) public key used in TPM Protocols and uniquely associated with each TPM device via an Infineon X509 Certificate; installed at the factory.

3 Roles, Authentication and Services

The TPM supports three roles, a CO role, a User role and a DUP role, as described in Table 12.

The TPM:

- Does not support a maintenance role or concurrent operators.
- Requires re-authentication following a power cycle.

Table 12: Roles Supported by the Module

Role ID	Role Description
CO	Cryptographic Officer, also known as the TPM Administrator or Admin Role. Controls certification of objects and changes Authentication Data of objects.
User	User, also known as the object owner. Uses the TPM to create cryptographic objects and to obtain cryptographic services for cryptographic objects.
DUP	Duplication Officer. Uses the TPM to duplicate TPM objects.

3.1 TPM Identification and Authentication Methods

The TPM supports the following Authentication Methods:

3.1.1 Password Verification

Operators in the CO or User roles are authenticated by a demonstration of knowledge of a Password as authentication data. Typically this will be used (but is not restricted) in a limited pre-boot environment.

3.1.2 HMAC Challenge-Response Authentication

This Challenge-Response Authentication is described as HMAC Authorization Session within TCG Specifications. Operators in the CO or User roles are authenticated by a challenge and response demonstration of knowledge of a shared secret. The shared secrets are HMAC-SHA1 and HMAC-SHA256 cryptographic keys. The TPM HMAC authorization session mechanism includes nonce values to prevent replay attacks.

3.1.3 Enhanced Authorization for Authentication

Enhanced Authorization is also referred to as Policy Authorization Session within TCG Specifications. Enhanced Authorization allows object creators to define specific actions and test which have to be performed before the service using the CSP key can be executed. The specific policy is encapsulated in a policy digest being a SHA-1/SHA-256 Digest Value and associated with the CSP key.

Operators in the CO or User roles can be authenticated via a policy digest, which requires as action the use of an authentication mechanism. Password Verification or HMAC Challenge-Response Authentication as described above, or Challenge-Response Authentication based on a Public Key Digital Signature Algorithm (RSA 2048-bit or ECDSA 256-bit) can be used as authentication mechanism. The TPM policy authorization session mechanism includes nonce values to prevent replay attacks. For guidance on the use of Enhanced Authorization for authentication please refer to Section 8.1.

3.1.4 Role Based Authentication Method Summary

The following table shows the allowed authentication mechanisms and data options for each Role. Enhanced Authorization is always allowed for each Role. For CO and USER Role the TPM object attributes control if Password and Challenge-response Mechanism is allowed in addition.

Table 13: Roles and Required Identification and Authentication

Role ID	Type/Mechanism of Authentication	Authentication Data
CO	Password verification	Password
	Challenge-response authentication using HMAC-SHA1 or HMAC-SHA256	Cryptographic Key (HMAC 160-bit key or HMAC 256-bit key)
	Enhanced Authorization requiring an authentication mechanism	Password or Cryptographic Key included as reference in the policy digest
User	Password verification	Password
	Challenge-response authentication using HMAC-SHA1/HMAC-SHA256	Cryptographic Key (HMAC 160-bit key or HMAC 256-bit key)
	Enhanced Authorization requiring an authentication mechanism	Password or Cryptographic Key included as reference in the policy digest
DUP	Enhanced Authorization requiring an authentication mechanism	Password or Cryptographic Key included as reference in the policy digest

3.1.5 Strength of Mechanism

Table 14: Strength of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
Password verification	When using the password verification mechanism, a password consisting of at least 12 alphanumeric characters shall be used, see Section 8.1 for details. Assuming as a worst case that the operator uses 12 decimal digits only, but still randomly chosen, the probability that a single random authentication attempt (by guessing the password value) will succeed is: $10^{-12} < 10^{-6}$ A very conservative estimate of the maximum authentication rate is 10^6/minute (60 µs per attempt). Under this assumption the probability that random authentication attempts will succeed within a one-minute interval is: $10^6 \times 10^{-12} = 10^{-6} < 10^{-5}$
Challenge-response authentication using HMAC	As a worst case it is assumed that for challenge-response authentication HMAC-SHA1 is used, which has smaller key length and smaller tag length (160 bit each) than HMAC-SHA256. Under this assumption the probability that a random authentication attempt (by guessing key value or tag value) will succeed is: $2^{-160} = 6.8\times10^{-49} < 10^{-6}$ With the same assumed maximum authentication rate of 10^6/minute as above, the probability that random authentication attempts will succeed within a one-minute interval is: $10^6 \times 6.8\times10^{-49} = 6.8\times10^{-43} < 10^{-5}$

Authentication Mechanism	Strength of Mechanism
Enhanced Authorization requiring an authentication mechanism	For the strength of this mechanism when using password verification or HMAC challenge-response authentication as required authentication mechanism see the two rows above. For challenge-response authentication using a Public Key Digital Signature Algorithm as required authentication mechanism it is assumed as worst case that RSA 2048-bit is used, which provides 112-bit security strength (ECDSA 256-bit provides 128-bit security strength). Therefore the probability that a random authentication attempt will succeed using this authentication mechanism is: $2^{-112} = 1.9\times10^{-34} < 10^{-6}$ With the same assumed maximum authentication rate of 10^6/minute as above, the probability that random authentication attempts will succeed within a one-minute interval is: $10^6 \times 1.9\times10^{-34} = 1.9\times10^{-28} < 10^{-5}$

3.2 Services

All services implemented by the TPM are listed in the tables below, with corresponding access to CSPs indicated according to the legend below. See [TPM_LIB] for a public description of all commands.

All Services are available in Full Operational Mode, Services available in Platform Initialization (PI) Mode are formatted bold and marked with index “PI”. For a detailed description of the Platform Initialization Mode please see section 1.1.

E = Execute: The TPM executes using the CSP.
G = Generate: The TPM generates the CSP.
R = Read: A CSP is output from the TPM.
B= Backup: TPM performs a Backup of a CSP.
W = Write: The TPM writes or updates the CSP.
Z = Zeroize: The module zeroizes the CSP.
-- = Not accessed by the service.
O = restore: The TPM restores the CSP.

Table 15: Unauthenticated Services CSP Access

Unauthenticated Services	DRBG-EI	DRBG-STATE	TPM-EPS	TPM-PS	TPM-Proof	AS-AD-K	AS-SALT	AS-SK	ES-KDK	ES-EK	KW-KDK	KW-IK	KW-CK	SIGK	IFX- PE-KEK	TPM-KEK	U-KEK	U-MACK	U-E-KAK	SIGK-PUB	TPM-KEK-PUB	U-KEK-PUB	U-E-KAK-PUB	IFX-SIGK-PUB	IFX-PE-KEK-PUB
TPM DRBG Services [1]																									
*** DRBG Generated Random Number [PI]**	--	EW	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
* DRBG Reseed	GEZ	EW	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
TPM Credential Protection																									
***** External Entity Authentication [2]	--	EW	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--	E
User Cryptographic Support Function																									
***** RSA Key Transport Scheme [2]	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--
* ECC CDH Primitive [2]	--	EW	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--
* Verify Digital Signature [RSA, ECC]	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--	--	--
*** Create HASH [PI]**	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
TPM Protected Storage [Public Keys Only]																									
* Write & Read Public Keys [4]	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	RW	RW	RW	--	--	R
TPM Enhanced Authorization																									
* Create+Verify HMAC Signature [3]	--	--	--	--	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
* Verify Digital Signature [RSA, ECC]	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--	--	--
TPM Field Upgrade Service																									
* Create HASH	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
TPM Startup																									
*** DRGB Instantiate [PI]**	GEZ	EW	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
*** Zeroize CSP [PI]**	--	--	--	--	--	Z	--	--	--	--	--	--	--	Z	Z	Z	Z	Z	--	Z	Z	Z	--	--	--
TPM Selftest & Show Status Service																									
* Self Test	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
* Show Status	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Unauthenticated Services Notes:

1) Generation of Random Numbers based on SP800-90A, allowed per IG3.1.
2) This service uses the Static Public Key of the relevant Key Establishment Method in the Module and does not disclose, modify, or create any CSPs, allowed per IG3.1.
3) This service is used for local TPM Data Structure Verification and Generation Services for the purpose of re-signing of data and does not disclose, modify or create any CSPs, allowed per IG3.1.

4) This service is only used for Public Key Entry and does not modify or substitute any CSP.

Utility Support services are unauthenticated services, which are always associated with a larger set of operations or services. These larger sets of operations or services comply with FIPS CSP Authentication Requirements.

Table 16: Utility Support Services CSP Access

Utility Support Services	DRBG-EI	DRBG-STATE	TPM-EPS	TPM-PS	TPM-Proof	AS-AD-K	AS-SALT	AS-SK	ES-KDK	ES-EK	KW-KDK	KW-IK	KW-CK	SIGK	IFX-PE-KEK	TPM-KEK	U-KEK	U-MACK	U-E-KAK	SIGK-PUB	TPM-KEK-PUB	U-KEK-PUB	U-E-KAK-PUB	IFX-SIGK-PUB	IFX-PE-KEK-PUB
TPM Context Management [1]																									
* Derive Keys based on KB KDF	--	--	--	--	--	--	--	--	--	--	E	--	G	--	--	--	--	--	--	--	--	--	--	--	--
* Wrap CSPs [AES and HMAC Protection]	--	--	--	--	--	--	--	--	--	--	--	E	EZ	--	--	--	--	--	--	--	--	--	--	--	--
* Backup CSP	--	--	--	--	--	B	--	B	--	B	--	--	--	B	B	B	B	B	--	--	--	--	--	--	--
* Restore CSP	--	--	--	--	--	O	--	O	--	O	--	--	--	O	O	O	O	O	--	--	--	--	--	--	--
* Zeroize CSP	--	--	--	--	--	Z	--	Z	--	Z	--	--	--	Z	Z	Z	Z	Z	--	--	--	--	--	--	--
TPM Session Service Initialization [2]																									
* RSA Key Transport Scheme	--	--	--	--	--	--	W	--	--	--	--	--	--	--	E	E	E	--	--	--	--	--	--	--	--
* ECC Key Agreement Scheme	--	--	--	--	--	--	G	--	--	--	--	--	--	--	E	E	E	--	--	--	--	--	--	WEZ	--	--
* Derive Keys based on KB KDF	--	--	--	--	--	E	EZ	GW	GEZ	GW	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Utility Support Service Notes:

1) CSPs can only be backed up if their associated set of operations (e.g. creation or loading of the CSP) complies to FIPS CSP Authentication Requirements. Restored CSPs can only be used by the associated operations (e.g. Signature Creation), which comply to FIPS CSP Authentication Requirements. Backup and Restore Services are protected via FIPS Approved Key Wrapping.

2) CSPs are only generated if the CSPs involved in the generation have been loaded via an Authenticated Service.

Table 17: User Authenticated Services CSP Access

User Authenticated Services	DRBG-EI	DRBG-STATE	TPM-EPS	TPM-PS	TPM-Proof	AS-AD-K	AS-SALT	AS-SK	ES-KDK	ES-EK	KW-KDK	KW-IK	KW-CK	SIGK	IFX-PE-KEK	TPM-KEK	U-KEK	U-MACK	U-E-KAK	SIGK-PUB	TPM-KEK-PUB	U-KEK-PUB	U-E-KAK-PUB	IFX-SIGK-PUB	IFX-PE-KEK-PUB
TPM Protected Storage																									
* Derive Primary Asym Key Pair (RSA, ECC)	--	--	E	E	--	--	--	--	--	--	--	--	--	G	--	G	G	--	--	--	--	--	--	--	--
* Derive Primary SymKeys (HMAC)	--	--	E	E	--	--	--	--	--	--	--	--	--	--	--	--	--	G	--	--	--	--	--	--	--
* Write Primary Sym Key (HMAC)	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	W	--	--	--	--	--	--	--
* Derive Primary KDK for Asym Storage Key	--	--	E	E	E	--	--	--	--	--	G	--	--	--	--	--	--	--	--	--	--	--	--	--	--
* Generate Asym Key (RSA,ECC)	--	EW	--	--	--	--	--	--	--	--	--	--	--	G	--	G	G	--	--	--	--	--	--	--	--
* Generate Sym Key (HMAC)	--	EW	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	G	--	--	--	--	--	--	--
* Write Sym Key (HMAC)	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	W	--	--	--	--	--	--	--
* Generate Sym KDK for Asym Storage Key	--	EW	--	--	--	--	--	--	--	--	G	--	--	--	--	--	--	--	--	--	--	--	--	--	--
* Derive Keys based on KB KDF	--	--	--	--	--	--	--	--	--	--	E	G	G	--	--	--	--	--	--	--	--	--	--	--	--
* Wrap CSPs [AES and HMAC Protection]	--	--	--	--	--	--	--	--	--	--	--	EZ	EZ	--	--	--	--	--	--	--	--	--	--	--	--
* Create HMAC Signature	--	--	--	--	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
* Write CSP	--	--	--	--	--	W	--	--	--	--	W	--	--	W	--	W	W	W	--	W	W	W	--	--	--
* Read CSP	--	--	--	--	--	R	--	--	--	--	R	--	--	R	--	R	R	R	--	R	R	R	--	--	R
TPM Hierarchy Management																									
* Generate CSP	--	EW	GW	GW	GW	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
* Zeroize CSP	--	--	Z	Z	Z	Z	--	--	--	--	--	--	--	Z	Z	Z	Z	Z	--	--	--	--	--	--	--
*** Write CSP [Authentication Data]** [PI]	--	--	--	--	--	W	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
TPM Import Services																									
* RSA Key Transport Scheme	--	--	--	--	--	--	--	--	--	--	W	--	--	--	E	E	--	--	--	--	--	--	--	--	--
* ECC Key Agreement Scheme	--	--	--	--	--	--	--	--	--	--	G	--	--	--	E	E	--	--	--	--	--	--	WEZ	--	--
* Derive Keys based on KB KDF	--	--	--	--	--	--	--	--	--	--	EZ	G	G	--	--	--	--	--	--	--	--	--	--	--	--
* Re-Wrap CSPs [AES and HMAC Protection]	--	--	--	--	--	--	--	--	--	--	--	EZ	EZ	--	--	--	--	--	--	--	--	--	--	--	--
* Read CSP	--	--	--	--	--	R	--	--	--	--	R	--	--	R	--	R	R	R	--	R	R	R	--	--	R
TPM Attestation																									
* Create Digital Signature [RSA, ECC]	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--	--	--	--	--	--	--	--	--
* Verify HMAC Signature	--	--	--	--	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
User Cryptographic Support Function																									
* RSA Key Transport Scheme	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--	--	--	--	--	--
* ECC CDH Primitive	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--	--	--	--	--	--
* Create Digital Signature [RSA, ECC]	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--	--	--	--	--	--	--	--	--
* Create HMAC	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--	--	--	--	--
*** Create HASH** [PI]	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
TPM Enchanced Authorization & Validation																									
* Create & Verify HMAC Signature	--	--	--	--	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

TPM Context Management																									
* Write CSP	--	--	--	--	--	W	--	--	--	--	W	--	--	W	W	W	W	W	--	W	W	W	--	--	W
* Zeroize CSP	--	--	--	--	--	Z	--	--	--	--	Z	--	--	Z	Z	Z	Z	Z	--	Z	Z	Z	--	--	Z

Table 18: ADMIN (CO) Authenticated Services CSP Access

ADMIN (CO) Authenticated Services	DRBG-EI	DRBG-STATE	TPM-EPS	TPM-PS	TPM-Proof	AS-AD-K	AS-SALT	AS-SK	ES-KDK	ES-EK	KW-KDK	KW-IK	KW-CK	SIGK	IFX-PE-KEK	TPM-KEK	U-KEK	U-MACK	U-E-KAK	SIGK-PUB	TPM-KEK-PUB	U-KEK-PUB	U-E-KAK-PUB	IFX-SIGK-PUB	IFX-PE-KEK-PUB
TPM Credential Protection																									
* External Entity Authentication	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	E	--	--	--	--	--	--	--	--	--
TPM Key Attestation																									
* Create Digital Signature [RSA, ECC]	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--	--	--	--	--	--	--	--	--	--	--
TPM Protected Storage Management																									
* Write CSP [Authentication Data]	--	--	--	--	--	W	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
TPM Field Upgrade Service																									
* Verify Digital Signature [RSA]	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	E	--

Table 19: DUP Authenticated Services CSP Access

DUP Authenticated Services	DRBG-EI	DRBG-STATE	TPM-EPS	TPM-PS	TPM-Proof	AS-AD-K	AS-SALT	AS-SK	ES-KDK	ES-EK	KW-KDK	KW-IK	KW-CK	SIGK	IFX-PE-KEK	TPM-KEK	U-KEK	U-MACK	U-E-KAK	SIGK-PUB	TPM-KEK-PUB	U-KEK-PUB	U-E-KAK-PUB	IFX-SIGK-PUB	IFX-PE-KEK-PUB
TPM Duplication																									
* RSA Key Transport Scheme	--	EW	--	--	--	--	--	--	--	--	G	--	--	--	--	--	--	--	--	--	E	--	--	--	E
* ECC Key Agreement Scheme	--	EW	--	--	--	--	--	--	--	--	G	--	--	--	--	--	--	--	GEZ	--	E	--	GRZ	--	E
* Derive Keys based on KB KDF	--	--	--	--	--	--	--	--	--	--	EZ	G	G	--	--	--	--	--	--	--	--	--	--	--	--
* Wrap CSPs [AES and HMAC Protection]	--	--	--	--	--	--	--	--	--	--	--	EZ	EZ	--	--	--	--	--	--	--	--	--	--	--	--
* Read CSPs	--	--	--	--	--	R	--	--	--	--	R	--	--	R	--	R	R	R	--	R	R	R	--	--	--

The following unauthenticated services are part of the authentication process and do not create, modify, disclose or substitute cryptographic keys or CSPs in accordance with IG 3.1.

Table 20: TPM Challenge + Response Authentication and Encryption Services

TPM Challenge+Response Authentication and Encryption Services	DRBG-EI	DRBG-STATE	TPM-EPS	TPM-PS	TPM-Proof	AS-AD-K	AS-SALT	AS-SK	ES-KDK	ES-EK	KW-KDK	KW-IK	KW-CK	SIGK	IFX-PE-KEK	TPM-KEK	U-KEK	U-MACK	U-E-KAK	SIGK-PUB	TPM-KEK-PUB	U-KEK-PUB	U-E-KAK-PUB	IFX-SIGK-PUB	IFX-PE-KEK-PUB
* Message Authentication using HMAC	--	--	--	--	--	E	--	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
* Message Enryption using AES Encrypt Decrypt	--	--	--	--	--	--	--	--	--	E	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
* Zeroize CSP	--	--	--	--	--	--	--	Z	--	Z	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

4 Self-tests

On power-on or reset, the TPM performs self-tests as described in Table 21 below. All KATs/PCT must be completed successfully prior to any other use of cryptography by the TPM. If one of the KATs/PCTs fails, the system is halted (in the Failure Mode state). In this mode, only TPM2_GetTestResult and TPM2_GetCapability is accepted by the TPM; no CSP access is possible. Self-tests may be invoked at any time using TPM2_SelfTest, with self-test results returned using TPM2_GetTestResult. All required Self-tests are performed before entering Full Operational Mode, Self-tests to be successfully performed before entering Platform Initialization (PI) Mode are formatted bold and marked with an index "PI". For a detailed description of the Platform Initialization Mode please refer to section 1.1.

Table 21: TPM Self-Tests

Self-Test	Description
Critical Function Self-Tests	
Hardware Integrity Test [PI)]	The TPM performs a hardware integrity test at power-up and at fixed periods. In either case, if the hardware integrity tests fails, TPM hardware immediately enters a security reset state (the TPM is mute).
Power-On Self-Tests	
Firmware Integrity [PI)]	SHA-256 message digest performed over all code located in NVM. This integrity test is not required or performed for code stored in masked ROM code memory.
DRBG KATs (Certs. #1886) [PI)]	Performs a fixed input KAT, inclusive of the SP 800-90A health monitoring tests.
SHA-1 KAT (Cert. #4129) [PI)]	Performs a fixed input KAT for SHA-1.
HMAC-SHA256 KAT (Cert. #3383) [PI)] **SHA-256 KAT** (Cert. #4129) [PI)]	Performs a fixed input KAT for HMAC-SHA256. By this also SHA-256 is known-answer tested.
AES KATs (Certs. #5069)	Performs separate encrypt and decrypt KATs using an AES-128 key in CFB mode.
RSA KATs (Certs. #2749)	Performs RSA Digital Signature (RSASSA-PKCS1-V1_5 using SHA256) Generation and Verification KATs using an RSA 2048-bit key.
ECDSA PCT (Certs. #1314)	Performs an ECDSA pairwise consistency test (PCT) using NIST recommended curve P-256.
ECC KAS KAT (Certs. #157)	Performs ECC CDH primitive Z computation using NIST-recommended curve P-256. The Single-Step KDF (SHA-1, SHA-256) Power-On Self-Test will be performed via testing the underlying SHA-1 and SHA-256 as described by IG 9.6.
KBKDF KAT (Certs. #172) HMAC-SHA1 KAT, HMAC-SHA256 KAT (Certs. #3383)	Performs two fixed input KAT of the SP 800-108 KDF using HMAC-SHA1 and HMAC-SHA256. By this also HMAC-SHA1 and HMAC-SHA256 is known-answer tested.
Conditional Self-Tests	
NDRNG CRNGT [PI)]	The TPM performs the AS09.42 to assure the NDRNG output is different than the previous value. Failure of this NDRNG CRNGT is treated as an attack; the TPM enters an error state. This conditional self-test is also performed in operational mode when the DRBG is reseeded (and therefore new entropy from the NDRNG is collected).
DRBG Health Monitoring Test [PI)]	Health tests on the CTR_DRBG Generate function is performed as required by NIST SP 800-90A.

Self-Test	Description
RSA Key Gen PCT	On generation of a RSA key pair, the TPM performs a pairwise consistency test. For key transport keys, the PCT sequence is encrypt/decrypt; for signature keys, the PCT sequence sign/verify is applied.
ECC Key Gen PCT	On generation of an ECC key pair, the TPM performs an ECDSA pairwise consistency test.
Firmware Load Test	The TPM performs RSA-2048 signature verification with SHA-256 over the firmware code to be loaded. New firmware is loaded only if verification succeeds.
Key load test	When RSA or ECC key pairs or public keys are loaded into TPM, TPM performs – depending on the key type – pair-wise consistency or key regeneration tests and/or assurance tests as required by NIST SP 800-56A and NIST SP 800-56B. Key material is loaded only if corresponding key load test succeeds.

5 Physical Security Policy

The TPM is a single-chip implementation that meets commercial-grade specifications for power, temperature, reliability, and shock/vibrations. The TPM employs standard passivation techniques. The TPM is intended for deployment on standard PCBs or similar assemblies. TPM packaging provides opacity and tamper evidence protections and will cause serious damage to the module, sufficient to meet FIPS 140-2 Physical Security Level 3.

TPM comes with a hard and opaque enclosure (see images in Section 1.2). Any attempt of physical tampering by mechanical means will leave evidence in form of scratches, broken edges of the enclosure or similar. All testing was performed at ambient temperature.

The TPM shall be visually inspected for evidence of tampering at least once before integration into a host device. After integration, it is possible to check for tamper evidence by opening the host device and inspecting the TPM.

6 Electromagnetic Interference and Compatibility (EMI/EMC)

The Module conforms to the EMI/EMC requirements specified by part 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B.

7 Mitigation of Other Attacks Policy

The TPM implements the mechanisms listed in Table 22 to mitigate attacks beyond the requirements of FIPS 140-2 Security Level 2. There are no specific limitations for any of these attack mitigations.

Table 22: Mitigation of Other Attacks

Other Attack	Mitigation Mechanism
Fault induction	External clock conditions, temperature and electromagnetic radiation (e.g. light) are monitored using sensors. Operation outside specific parameters causes the chip to enter the *Security reset* state until the condition is cleared.
Software fault induction	The virtual physical address mapping together with the memory management unit (MMU) gives the possibility to define different access rights for memory areas. In case of an access violation (e.g., embedded software trying to read memory of IC-dedicated software) hardware enters the *Security reset* state.
EEPROM memory corruption	The memory system maintains EEPROM data integrity using an error detection and correction mechanism at the hardware level. A 1-bit (per byte) error is automatically corrected; multiple-bit errors cause the TPM to enter the *Security reset* state.
Design analysis and surveillance attacks (in operational or power off conditions)	The TPM integrated circuit level layout uses masking, critical circuit shielding and synthesized logic to deter attacker knowledge of the part design. Outer layer lines are protected with a proprietary masking technique, with active shielding in internal layers to protect the masking mechanism. The use of synthesized logic deters attackers from pattern recognition of logic clusters. As well, a dedicated CPU with a non-public bus protocol is used which makes analysis complicated.
Physical probing of memory and data buses	Proprietary memory and bus masking to deter probing memories or buses.

8 Security Rules and Guidance

The TPM implementation also enforces the following security rules:

- No additional interface or service is implemented by the Module which would provide access to CSPs.
- Data output is inhibited during key generation, self-tests, zeroization, and error states.
- There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
- The module does not support manual key entry.
- The module does not output plaintext CSPs or intermediate key values.
- Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.

8.1 Requirements for Secure Operation

The application must assure the following conditions are met for operation of the TPM in the FIPS 140-2 Approved mode:

Requirements for Approved and Allowed Function Usage:

- Only Approved and allowed cryptographic functions as listed in Table 7 and Table 8 may be used.
- Non-approved cryptographic functions as listed in Table 9 shall not be used.

Requirements for Key Entry and Output to and from the Module:

- When entering CSP keys into the module the operator shall ensure usage of FIPS Approved Key Wrapping via usage of Message Encryption using AES Encrypt Decrypt in combination with Message Authentication using HMAC Service listed in Table 20.
- When generating CSP keys for duplication (export) the operator shall set the CSP key attribute (encryptedDuplication), which enforces FIPS Approved key transport/ agreement and wrapping during export.
- When importing CSP keys into the module the operator shall only import CSP keys, which attributes (encryptedDuplication) requires FIPS Approved key transport/ agreement and wrapping.
- Unauthorized loading of Secret Keys into the module via TPM2_LoadExternal shall not be used.

Requirements for Authentication:

- When using the password verification mechanism for operator authentication, a password consisting of at least 12 randomly chosen characters, containing at least one character of the following 4 groups: uppercase letters, lowercase letters, numerals and symbols but still randomly chosen shall be used.
- When using Enhanced Authorization the operator shall ensure that the policy will require at least one of the following authentication mechanisms:
  - Password verification
  - Challenge-response mechanism based on a Message Authentication Code
  - Challenge-response mechanism based on a Public Key Digital Signature Algorithm

Requirements for Initialization

- In case the TPM is in an un-owned state (e.g. default state after shipment) the operator shall initialize the authentication data to control the Owner and the Endorsement Hierarchy via usage of the *Write CSP Service [Authentication Data]* listed in the TPM Hierarchy Group in Table 17. For detailed information on how to perform this please refer to section 9.
- After each Reset the operator shall initialize the authentication data to control the Platform Hierarchy via usage of the *Write CSP Service [Authentication Data]* listed in the TPM Hierarchy Group in Table 17.

9 Annex A – Module Initialization

When the TPM 2.0 is in an un-owned state (e.g. when the TPM is in the default state after shipment) the TPM shall be initialized, since all authentication values have a default value set to an EmptyAuth.

Module Initialization (also referred to as Taking Ownership of the TPM 2.0) basically means to initialize several authorization and policy values and optionally to create a primary storage key. Initializing the authorization values (endorsementAuth, ownerAuth, lockoutAuth) will be performed with TPM2_HierarchyChangeAuth and initializing the policies (endorsementPolicy, ownerPolicy, lockoutPolicy) will be done with TPM2_SetPrimaryPolicy.

The following flow shows a secure way to initialize the module:

- Check capabilities to see if ownership is enabled
  - TPM2_GetCapability with TPM_PT_PERMANENT and TPM_PT_STARTUP_CLEAR checking for ownerAuthSet == 0 and shEnable == 1
- TPM2_CreatePrimary() to get the IFX-PE-KEK, for which a certificate exists
- Check the Endorsement Key certificate to verify that ownership is taken of an authentic Infineon TPM
- Start an encrypted authorization session using the IFX-PE-KEK-PUB to protect the secret
- TPM2_HierarchyChangeAuth() using parameter encryption to protect the new auth values (endorsementAuth, ownerAuth, lockoutAuth)
- TPM2_SetPrimaryPolicy using the previously set auth values to set the corresponding policies (endorsementPolicy, ownerPolicy and lockoutPolicy)
- Optional: TPM2_CreatePrimary() to create a storage primary key
- Optional: TPM2_EvictControl to make the storage primary key persistent

10 Annex B – Module Startup

When the TPM 2.0 is reset (e.g. when power is supplied to the TPM) the TPM shall be correctly started up since some authentication values have a default value set to an EmptyAuth.

The TPM 2.0 sets the platformAuth to an EmptyAuth after a TPM reset (_TPM_Init) by default, which can be easily satisfied using the NULL password. To avoid control of the TPM 2.0 Platform Hierarchy using platformAuth from other entities than the platform BIOS, it is required to change the platformAuth to a secure random value immediately after a TPM reset. The changed platformAuth may be stored at a secure storage location, if needed by the BIOS or other processes during the platform boot. Before transitioning to the OS the platformAuth value must be securely discarded at the stored secure location (e.g. overwritten with zeroes).

The following flow shows a secure way to startup the module:

- Check the Endorsement Key certificate to verify that ownership is taken of an authentic Infineon TPM
- Start an encrypted authorization session using the IFX-PE-KEK-PUB to protect the secret
- TPM2_HierarchyChangeAuth using parameter encryption to protect the new auth values (platformAuth)
- TPM2_SetPrimaryPolicy using the previously set auth values to set the corresponding policies (platformPolicy)