STMICROELECTRONICS

Trusted Platform Module ST33TPHF2ESPI

ST33HTPH2E28AHA5 / ST33HTPH2E32AHA5 / ST33HTPH2E28AAE5 / ST33HTPH2E32AAE5

FIPS 140-2 Security Policy
Level 1

Firmware revision: 47.08
HW version: ST33HTPH revision A

Date: 2016/11/02
Document Version: 01-13

NON-PROPRIETARY DOCUMENT

Table of Contents

1 MODULE DESCRIPTION

1.1 Definition

The Trusted Platform Module ST33TPHF2ESPI is a fully integrated security module designed to be integrated into personal computers and other embedded systems. The security module is used primarily for cryptographic key generation, key storage and key management as well as generation and secure storage for digital certificates.

The TPM is a single chip cryptographic HW module as defined in **[FIPS 140-2]**. The single silicon chip is encapsulated in a hard, opaque, production grade integrated circuit (IC) package.

The cryptographic boundary is defined as the perimeter of the IC package. The security module supports an SPI interface compliant with the Trusted Computing Group (TCG) specification for PC Client interface [TIS 1.30]. The HW and FW cryptographic boundaries are indicated in §1.3.

The security module implements version 1.2 and the version 2.0 of the Trusted Computing Group (TCG) specification for Trusted Platform Modules (TPM). The TPM FW version 2.0 is excluded from the security requirements of FIPS 140-2 (please refer to §1.6).

1.2 Module identification

The hardware and firmware versions covered by the FIPS evaluation are identified as follow:

- Hardware version: ST33HTPH revision A
- Firmware version: 47.08

FW version can be retrieved through the command TPM_GetCapability: 1.2.47.08

The cryptographic services are provided by the cryptographic library “NesLib 4.2.9 for ST33”.

The product is manufactured in two packages:

- TSSOP28
  - TSSOP 28-pin
  - 4.4 x 9.7 mm
- VQFN32
  - Very thin pitch Quad pack no-lead 32-pin
  - 5 x 5 mm

Next pictures illustrate the 2 available packages for the security module:

Figure 1: Picture of cryptographic Module in VQFN32 package (AE5 configuration)

Figure 2: Picture of cryptographic Module in TSSOP28 package (HA5 configuration)

Those two packages are available in 2 different configurations of the security module:

1.2.1 *P68HAHA5 configuration*

The default FW version of this configuration is 47.04. To operate with FW version 47.08, module FW must be first field upgraded from 47.04 to 47.08.

Table 1: Security module configuration – Marking P68HAHA5

	Module configuration	
Product name / HW version	ST33TPHF2ESPI/ ST33HTPH revision A	
Package	TSSOP28	VQFN32
Part number	ST33HTPH2E28AHA5	ST33HTPH2E32AHA5
Marking	P68HAHA5	
FW version	47.08	
T° range	-40°C to +105°C	

1.2.2 *P68HAAE5 configuration*

The default FW version of this configuration is 47.00. To operate with FW version 47.08, module FW must be first field upgraded from 47.00 to 47.08.

Table 2: Security module configuration – Marking P68HAAE5

	Module configuration	
Product name / HW version	ST33TPHF2ESPI/ ST33HTPH revision A	
Package	TSSOP28	VQFN32
Part number	ST33HTPH2E28AAE5	ST33HTPH2E32AAE5
Marking	P68HAAE5	
FW version	47.08	
T° range	-40°C to +105°C	

The pin layouts for the ST33TPHF2ESPI are shown in Figure 3 and Figure 4.

Figure 3: TSSOP28 Pinout Diagram

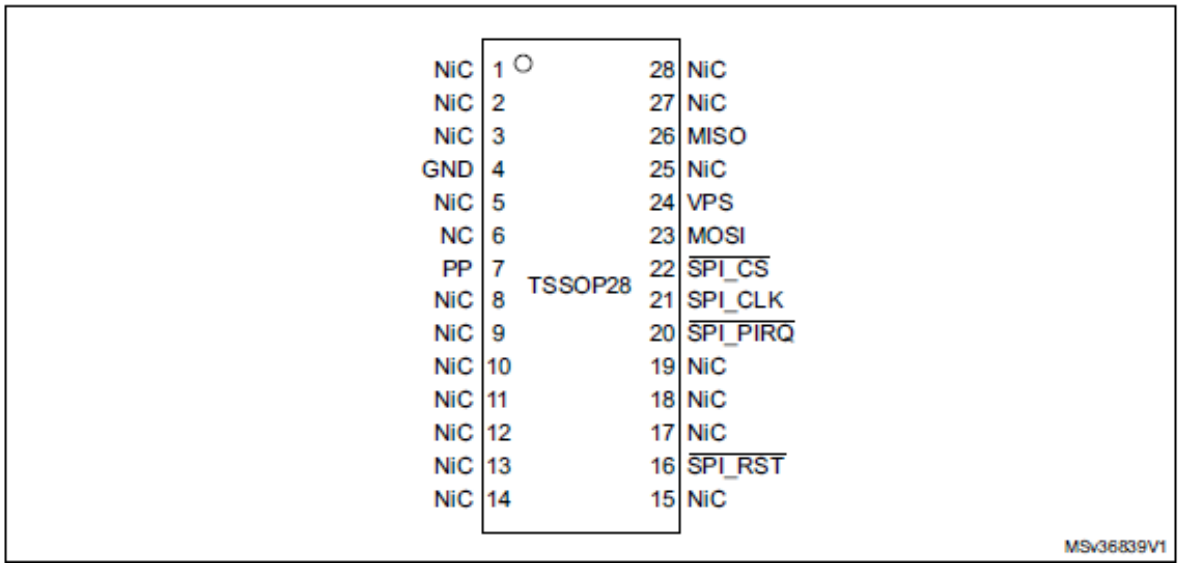
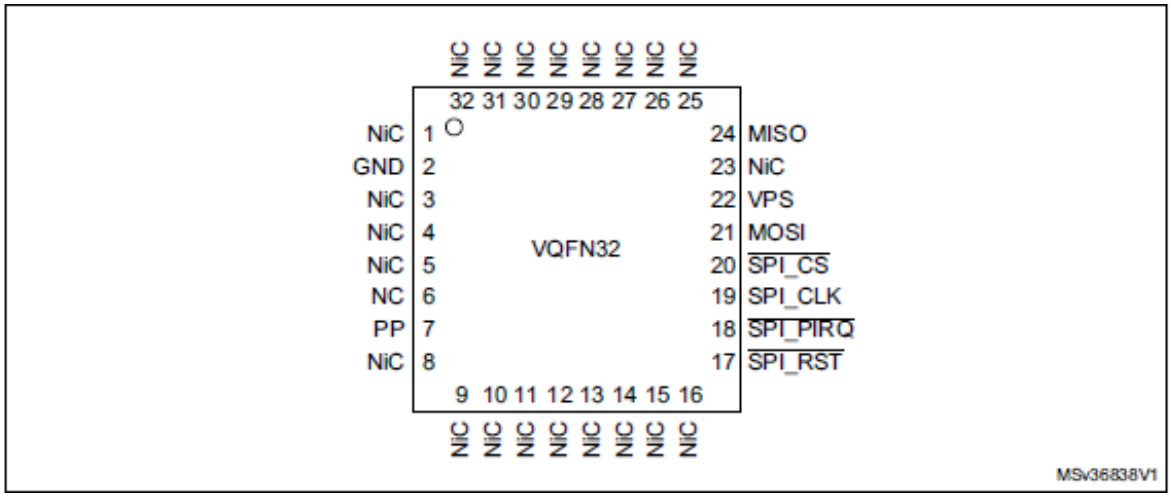


Figure 4: VQFN32 Pinout Diagram



Next table gives a description of the products pins.

Table 3: ST33TPHF2ESPI Pin definition

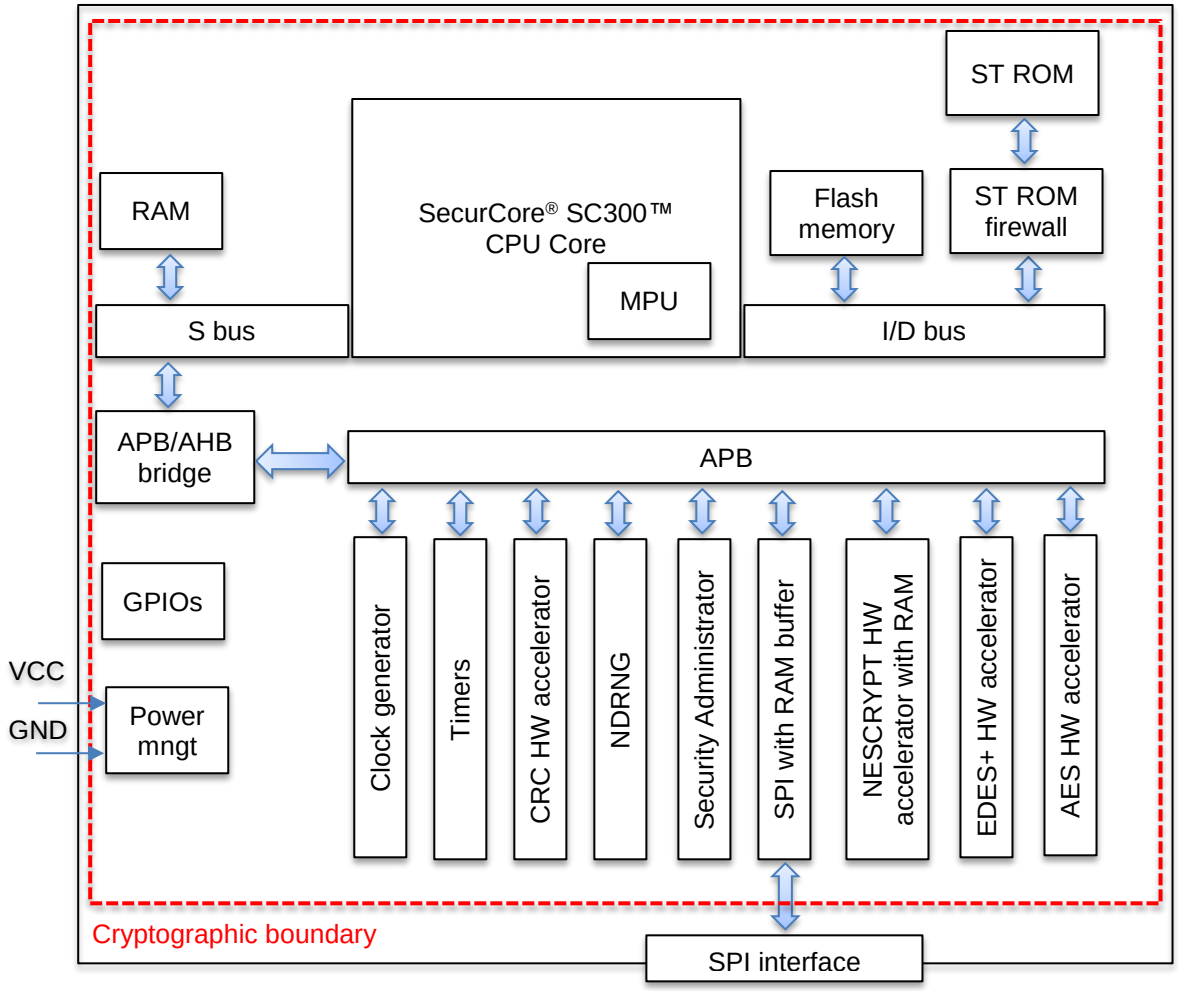
Signal	Type	Description
VPS	Input	**Power supply**. This pin must be connected to 1.8V or 3.3V DC power rail supplied by the motherboard.
GND	Input	GND has to be connected to the main motherboard ground.
$\overline{\text{SPI\_RST}}$	Input	**SPI Reset** used to re-initialize the device
MISO	Output	**SPI** Master Input, Slave Output (output from slave)
MOSI	Input	**SPI** Master Output, Slave Input (output from master)
SPI_CLK	Input	**SPI** serial clock (output from master)
$\overline{\text{SPI\_CS}}$	Input	**SPI** slave select (active low; output from master)
$\overline{\text{SPI\_PIRQ}}$	Output	**SPI IRQ** used by TPM to generate an interrupt
PP	Input	**Physical presence**, active high, internal pull-down. Used to indicate Physical Presence to the TPM.
NiC	-	**Not internally connected:** not connected to the die. May be left unconnected but no impact on TPM if connected.
NC	-	**Not Connected:** connected to the die but not usable. May be left unconnected. Internal pull-down.

1.3 Block diagrams

A logical block diagram of the hardware ST33HTPH is provided at Figure 5: ST33HTPH block diagram. TPM is composed of:

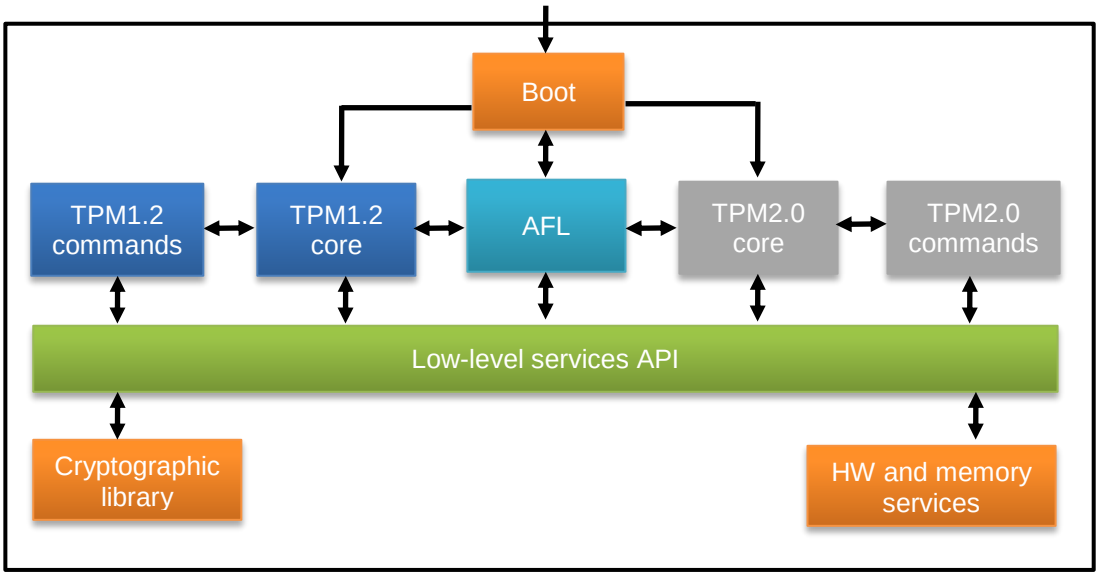
- A SecurCore® SC300™ CPU core including a MPU (Memory Protection Unit)
- Memories (RAMs, Flash and ROM)
- HW accelerators for CRC (16 and 32-bits) and cryptographic operations (symmetric with EDES+ and AES and asymmetric with NESCRYPT)
- A clock generator and three 16-bit timers
- NDRNG (non-deterministic random bit generator)
- SPI master/slave block
- A security administration block dedicated to chip security configuration and alarms detection
- FW and data stored in the memory areas

Figure 5: ST33HTPH block diagram



A block diagram of the TPM FW is provided in Figure 6: TPM FW block diagram.

Figure 6: TPM FW block diagram



TPM FW is composed of:

- Non-upgradable code blocks located in ROM & flash memories (depicted in orange)
  - Boot code
  - Cryptographic library
  - HW and memory services
- Upgradable code blocks via secure field upgrade mechanism (blue, grey and green boxes)
  - Application flash loader (AFL) in charge of TPM field upgrade
  - TPM1.2 core
  - TPM1.2 commands code
  - TPM2.0 core
  - TPM2.0 commands code
  - Low-level services API (incl. cryptographic services, memory management, …)

TPM2.0 core and TPM2.0 commands FW are irreversibly deactivated as indicated in §1.6.4 and are not part of the FIPS 140-2 evaluation.

1.4 Security levels

The cryptographic module meets the overall requirements applicable to Level 1 security of FIPS 140-2.

Table 4: Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services and Authentication	1
Finite State Model	1
Physical Security	1
Operational Environment	N/A
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	1
Overall	**1**

1.5 Cryptographic functions

The security module supports the following cryptographic algorithms (both approved and non-approved). Algorithm certificate numbers for each approved algorithm are listed below.

Table 5: Cryptographic Functions

Algorithm		Approved	Certificate number
RSA	Digital Signature Verification with key length = 1024 bits	Legacy use	#2057
	Digital Signature Verification with key length = 2048 bits	Yes	
	Digital Signature Generation with key length = 2048 bits		
	Key generation with key length = 2048 bits		
	Digital Signature Generation with 1024 ≤ key length <2048 bits	No	NA
	Key generation with key length ≤ 1024 bits		
	Key wrapping with key length = 2048 bits	Allowed	NA
Secure SHA-1[1]	Digital Signature Verification	Legacy use	#3306
	Non-digital signature generation applications	Yes	
	Digital signatures generation	No	NA
Non secure SHA-1	Digital Signature Verification	Legacy use	#3305
	Non-digital signature generation applications	Yes	
	Digital signatures generation	No	NA
SHA-256	Digital Signature Verification	Yes	#3306
	Non-digital signature generation applications		
	Digital signatures generation		
	DRBG		#3305
HMAC SHA-1[2]	key length >=160 bits	Yes	#2614
AES (CTR and CFB modes with key length = 128bits)		Yes	#4001
KDF 800-135 (TPM) [3]		Yes	#829
DRBG 800-90A (Hash_DRBG)		Yes	#1191
KDF 800-108 (Counter mode)		Yes	#93
KTS (AES cert #4001 + HMAC cert #2614) for key transport		Yes	NA
MGF1		No	NA
NDRNG (True random number generator) used to: • Seed or reseed DRBG 800-90A (with approximatively 366 bits of entropy) • Generate random numbers not dedicated to be used as cryptographic material		Allowed	NA

[1] Fault injection resistant algorithm
[2] HMAC SHA-256 is not used by the TPM
[3] TPM key establishement protocol that uses TPM KDF has not been reviewed or tested by the CAVP and CMVP (IG D.11)

1.6 Mode of Operation

This security policy only applies to the security module when it is configured:

- In FIPS mode and when this mode is irreversibly locked
- In TPM1.2 mode and when this mode is irreversibly locked

1.6.1 FIPS activation

The FIPS mode can be configured via:

- TPM_SetCapability command (capability = PERMANENT FLAGS, subCap = FIPS)
- TPM_SetMode proprietary command (mode = TPMFips) with TPM owner authorization or Physical presence

To irreversibly lock the FIPS mode, both following operations must be done:

- NV area must be locked via TPM_NV_DefineSpace (nvIndex = TPM_NV_INDEX_LOCK)
- FIPS flag lock must be set via TPM_SetMode (modeLock = TPMFipsLock)

In order to not reuse in FIPS mode, keys and CSPs generated in non FIPS mode, TPM_ForceClear command (before or after FIPS activation) or TPM_OwnerClear command (after FIPS activation) must be executed.

1.6.2 TPM1.2 mode lock

The TPM1.2 mode can be irreversibly locked via:

- TPM_SetMode proprietary command (mode = TPMLibLock) with TPM owner authorization or Physical presence

1.6.3 Verification

The FIPS mode status may be retrieved with the command TPM_GetCapability with the capability to TPM_PERMANENT_FLAGS.FIPS flag.

The FIPSLock flag may be retrieved with the command TPM_GetCapability with the capability set to TPM_CAP_MFR that provides the flag *TPMFipsLock* in the bitmap *modeLock*.

1.6.4 *FIPS mode guidance*

When FIPS mode is activated, TPM implementation:

- Prevents:
  - Generation, loading and import of RSA 1024-bit keys
- Does not prevent:
  - Usage of SHA-1 hash during digital signature generation
  - Usage of MGF-1 in some specific commands

To use TPM in a full approved FIPS 140-2 mode, TPM user:

- Shall use TPM_OSAP for authentication sessions with TPM_ET_AES128_CTR ADIP encryption scheme for commands listed in Table 11 : Encrypted methods for secret and private keys input and marked as using AES_CTR to input or output CSPs.
- Shall use SHA-256 hash algorithm for digital signature generation. It concerns the following services:
  - TPM_Sign
- Shall not use services that don't meet FIPS 140-2 criteria:
  - TPM_DAA_Join (use of MGF1 as encryption scheme)
  - TPM_DAA_Sign (use of MGF1 as encryption scheme)
  - TPM_CertifyKey (signature generation using SHA1)
  - TPM_CertifyKey2 (signature generation using SHA1)
  - TPM_Quote (signature generation using SHA1)
  - TPM_Quote2 (signature generation using SHA1)
  - TPM_TickStampBlob (signature generation using SHA1)
  - TPM_ReleaseTransportSigned (signature generation using SHA1)

1.7 Ports and interfaces

The physical port of the security module is the SPI Bus.

The logical interfaces and their mapping to physical ports of the module are described below:

Table 6 : Ports and interfaces

Logical interface	Description	Physical port
Control Input Interface	Control Input commands issued to the security module	$\overline{\text{SPI\_CS}}$ / SPI_CLK / MOSI / SPI_RST / PP
Status Output Interface	Status data output by the chip	$\overline{\text{SPI\_CS}}$ / SPI_CLK / MISO / SPI_PIRQ
Data Input Interface	Data provided to the chip as part of the data processing commands	$\overline{\text{SPI\_CS}}$ / SPI_CLK / MOSI
Data Output Interface	Data output by the chip as part of the data processing command	$\overline{\text{SPI\_CS}}$ / SPI_CLK / MISO
Power interface	Power interface of the chip	VPS / GND

Here are some details concerning the ports and interfaces of TPM:

1. The module does not include a maintenance interface.
2. Control and data inputs are multiplexed over the same physical interface (SPI bus). Control and data are distinguished by properly parsing input TPM command parameters according to input structures description, indicated for each command in **[TPM Part3 r116]**[1].
3. Status and data output are multiplexed over the same physical interface (SPI bus). Status and data are distinguished by properly setting output TPM response parameters according to output structures description, indicated for each command in **[TPM Part3 r116]**.
4. The logical state machine and the command structure parsing of the module prevent from using input data externally from the “data input path” and prevent from outputting data externally from the “data output path”.
5. While performing key generation or key zeroization (no manual key entry on TPM), the output data path is logically disconnected while the output status path remains connected to report any possible failure during command processing. Generally, the output data path is only connected when TPM outputs response containing data.
6. Plaintext data can be output through usage of:
   - TPM_UnBind
   - TPM_Unseal

   To prevent inadvertent release of the plaintext data, both commands performs:
   - Check of command input structure
   - Check of command authorization (cf. §2.2 for details)
   - Decryption of the input blob with private part of specified key
7. The logical state machine and command structure of the module guarantees the inhibition of all data output via the data output interface whenever an error state exists and while doing self-tests.

[1] Some commands only deal with control input and status output parameters

2 IDENTIFICATION AND AUTHENTICATION POLICY

This chapter gives details about the roles managed by TPM.

2.1 Roles

Services (services are listed in §3.2) proposed by TPM are accessible under different roles. Next table defines the different roles supported by the TPM.

Table 7 - Roles

Role	Description	Type of authentication	Authentication data
Crypto officer (CO)	Equivalent to TPM owner (cf. **[TPM Part1 r116]** for role definition). Some TPM services are reserved to owner (initialization/configuration).	Role based	160-bit secret data (Owner AuthData)
User (U)	Role requiring entity authorization, operator authorization.	Role based	160-bit secret data (key usageAuth or operator AuthData)
Physical presence (PP)	HW assertion that proves that an operator is physically present (no remote access)	Role based (HW based)	None
No authentication (NA)	Some TPM services do not require any authentication.	None	None

The security module does NOT provide a Maintenance Role or Maintenance Interface.

Cryptographic module does NOT support concurrent operators.

2.2 Authentication

2.2.1 Description

Crypto officer and user authentication data knowledge must be proven to authorize some TPM services. TPM uses a two-step mechanism for authorization that consists in:

1. Opening a session of the following types:
    a. OIAP: Object-Independent Authorization Protocol
    b. OSAP: Object-Specific Authorization Protocol
    c. DSAP: Delegation-Specific Authorization Protocol

    Session is used to establish a sequence of nonce-data included in the authorization process (protection against replay attacks). OSAP and DSAP sessions also create a shared secret used as HMAC key for command authorization. For OIAP, the authorization data is directly used as HMAC key.
2. Using the command to be authorized by verifying if HMAC (based on authorization value) passed as parameter corresponds to the value computed by TPM. If they match, command execution is authorized.

Secret authorization data is never exposed in plaintext (there is one exception for operatorAuth entered by TPM_SetOperatorAuth service and used by TPM_SetTempDeactivated). HMAC computation output based on the authorization data enables to prove knowledge of this secret.

When power is removed from the module, all existing authentication sessions are destroyed. Therefore, the module must re-authenticate every role or identity after each power-on sequence.

2.2.2 *Authorization strength*

As authorization values are 160-bit random values (based on unbiased distribution of '0' and '1'), the probability for an attacker to guess the authorization data is:

$$\frac{1}{2^{160}} = 6.84 * 10^{-49}$$

This value matches the requirement of $1*10^{-6}$ indicated in **[FIPS 140-2]**.

The number of attempts per minute that an attacker can make is limited by the DAM (Dictionary Attack Mechanism). DAM consists in counting the number of failed authentication. When this counter reaches a pre-defined threshold, a lockout period is started. During this period, no authorized command execution is allowed and a specific error (TPM_DEFEND_LOCK_RUNNING) is returned in TPM response until period expires. Next table indicates the threshold values and the lockout durations:

Table 8 : DAM lockout durations

Failed authentication counter	<10	10 (DAM threshold)	11	12	13	…	23	>23
Lockout period (in seconds)	0	10	20	40	60	…	81920	86400

This table indicates that an attacker can do a maximum (during the first minute) of 12 trials per minute (if failed authorization counter reaches 12 it means total lockout period is equal to 10s + 20s + 40s = 70s). As a result the probability per minute that a random attempt will lead to a successful authorization matches FIPS requirements. Value is equal to:

$$12 * \frac{1}{2^{160}} = 8.21 * 10^{-48}$$

This value matches the requirement of $1*10^{-5}$ indicated in **[FIPS 140-2]**.

NB: commands handling (reception, processing and response sending) is negligible compared to the lockout periods and not taken into account in the above computation.

3 ACCESS CONTROL POLICY

This chapter gives details about the services, keys and CSPs of the TPM.

3.1 List of Keys and CSPs

Table 9: Keys and CSPs list

Keys/CSPs		Description	Zeroization
Index	**Name**		
1	Endorsement key (EK) – private part	2048-bits permanent RSA key unique per TPM stored in the form of two prime numbers. EK primes are generated externally by a HSM and inserted during TPM production phase. EK is used to: • Decrypt encOwnerAuth and encSrkAuth in TPM_TakeOwnership command • Decrypt blob in TPM_ActivateIdentity	No zeroization (NIST waiver)
2	Storage root key (SRK) – private part & authorization value	2048-bits non-volatile RSA key. Root key of the key storage hierarchy. Key is generated and stored on TPM on TPM_TakeOwnership command according to the input parameters. SRK is used to: • Wrap and unwrap keys stored in the protected storage hierarchy Authorization data (non-volatile data) are 160-bits secret data used for SRK authorization. It is passed encrypted (RSA OAEP SHA1 algorithm with key = public part of EK) to TPM_TakeOwnership command. It is used as key for TPM KDF SP800-135 in session shared secret (CSP #8) generation for TPM_MakeIdentity and might be used for commands with U role in Table 10: Command support table that uses SRK as parent key.	TPM_OwnerClear TPM_ForceClear
3	User RSA keys – private part & authorization value	2048-bits RSA keys generated with TPM_CreateWrapKey, TPM_MakeIdentity and TPM_CMK_CreateKey commands (output encrypted from TPM with parent key indicated in the command). Keys loaded on the TPM via TPM_ActivateIdentity, TPM_LoadKey or TPM_LoadKey2. Depending on key attributes (keyUsage field in TPM_KEY structure), key can be used as: • Signing key (TPM_KEY_SIGNING) • Storage key (TPM_KEY_STORAGE) • Identity key (TPM_KEY_IDENTITY) • Binding key (TPM_KEY_BIND) • Signing and binding key (TPM_KEY_LEGACY) • Migration key (TPM_KEY_MIGRATE) Key might be volatile or non-volatile (keyFlags parameter in TPM_KEY structure).	TPM_OwnerClear TPM_ForceClear TPM_FlushSpecific TPM_EvictKey TPM_Init (for volatile keys only)

Keys/CSPs		Description	Zeroization
Index	Name		
		Authorization data (non-volatile data) are 160-bits secret data used for user RSA key authorization. It is passed encrypted (RSA OAEP SHA1 algorithm with key = public part of parent key) to key creation commands (TPM_CreateWrapKey, TPM_MakeIdentity and TPM_CMK_CreateKey). It is used as key for TPM KDF SP800-135 in session shared secret (CSP #8) generation for commands with U role in Table 10: Command support table that might use user RSA key as parent key.	
4	Field upgrade verification key	2048-bits permanent RSA key unique per TPM product line. Only public part of the key is stored in the TPM (modulus, exponent).	No (public key only)
5	contextKey / delegateKey	128-bits non-volatile AES key used to perform context saves/restores (TPM_SaveContext, TPM_LoadContext) and delegation blobs encryption/decryption (TPM_Delegate_CreateKeyDelegation, TPM_Delegate_CreateOwnerDelegation, TPM_Delegate_LoadOwnerDelegation). Key is generated by HDRBG on TPM_TakeOwnership command.	TPM_OwnerClear TPM_ForceClear
6	HDRBG input seed	48-bytes value output from a NDRNG.	Transient value
8	Session shared secret	160-bit volatile shared secret generated on TPM_OSAP or TPM_DSAP commands execution by derivation (TPM KDF SP800-135) using entity authorization data as key. Session shared secret is used as: • AES CTR key (first 128-bits) in ADIP protocol to encrypt/decrypt authorization data (list of commands is indicated in Table 11 : Encrypted methods for secret and private keys input). • HMAC SHA-1 key in HMAC computation in authorization protocols (concerned commands are indicated with CO or U role in Table 10: Command support table).	TPM_FlushSpecific TPM_OwnerClear TPM_ForceClear
9	NV index – authorization value	160-bits (non-volatile data) used as secret authorization data for a specific NV index. Value is passed encrypted (AES CTR 128 with key = OSAP shared secret) to the TPM_NV_DefineSpace command. It is used as key for TPM KDF SP800-135 in session shared secret (CSP #8) generation for TPM_NV_WriteValueAuth and TPM_NV_ReadValueAuth commands.	TPM_OwnerClear TPM_ForceClear
10	HDRBG state	222-bytes (volatile data) representing the HDRBG internal state (V and C secret values). HDRBG is seeded after each reset with NDRNG output (CSP #6). Internal state is updated after each HDRBG generate command execution or reseed. HDRBG is used in random number generation for cryptographic material.	TPM_OwnerClear TPM_ForceClear TPM_SetMode TPM_Init
11	tpmProof	160-bits secret random number (non-volatile data) generated by HDRBG on TPM_TakeOwnership command execution. It is used as: • HMAC SHA-1 key in integrity computation of blobs generated or read in the following commands:	TPM_OwnerClear TPM_ForceClear

Keys/CSPs		Description	Zeroization
Index	Name		
		TPM_CertifyKey2, TPM_Delegate_CreateKeyDelegation, TPM_Delegate_CreateOwnerDelegation, TPM_Delegate_UpdateVerification, TPM_CreateMigrationBlob, TPM_AuthorizeMigrationKey, TPM_CMKApproveMA, TPM_CMK_CreateKey, TPM_CMK_CreateTicket, TPM_CMK_CreateBlob, TPM_CMK_ConvertMigration, TPM_SaveContext, TPM_LoadContext, TPM_Seal	
12	Owner – authorization value	160-bits secret authorization data (non-volatile data) for owner authorization. It is passed encrypted (RSA OAEP SHA1 algorithm with key = public part of EK) to TPM_TakeOwnership command. It can be changed on TPM_ChangeAuthOwner command processing. It is used as key for TPM KDF SP800-135 in session shared secret (CSP #8) generation for all commands listed in Table 10: Command support table and requesting CO role to be authorized.	TPM_OwnerClear TPM_ForceClear
13	Monotonic counters – authorization value	160-bits secret authorization data (non-volatile data) for a specific monotonic counter (up to 4 monotonic counters can be created). Value is passed encrypted (AES CTR 128 with key = OSAP shared secret) to the TPM_CreateCounter command. It is used as key for TPM KDF SP800-135 in session shared secret (CSP #8) generation for TPM_IncrementCounter and TPM_ReleaseCounter commands.	TPM_ReleaseCounter TPM_ReleaseCounterOwner TPM_OwnerClear TPM_ForceClear
14	Pre-computed RSA keys – private part	2048-bits RSA keys (exponent = 65537) pre-computed during TPM background processing (between commands handling) and forming a pool of keys used to speed up key creation commands. Keys are non-volatile data.	TPM_OwnerClear TPM_ForceClear TPM_SetMode
15	Operator – authorization value	160-bits secret authorization data (non-volatile data) entered in plaintext on TPM_SetOperatorAuth. It is used as key for TPM KDF SP800-135 to be able to deactivate the TPM until the next boot of the platform via TPM_SetTempDeactivated command.	TPM_OwnerClear TPM_ForceClear

3.2 Services

Next table lists all services supported by the TPM in FIPS approved mode and indicates for each service, the role that can use this service and the keys/CSPs that can be accessed.

Table 10: Command support table

Services		Role	Keys and CSP access (R = read, W = write, O = output, Z = zeroize)
Admin Start up and State			
1	TPM_Init	**NA**	W: 1, 11 (first power-up only) Z: 3, 10
2	TPM_Startup	**NA**	-
3	TPM_SaveState	**NA**	-
Admin Testing			
4	TPM_SelfTestFull	**NA**	-
5	TPM_ContinueSelfTest	**NA**	-
6	TPM_GetTestResult	**NA**	-
Admin Opt-in			
7	TPM_SetOwnerInstall	**PP**	-
8	TPM_OwnerSetDisable	**CO**	R: 6, 8, 10 W: 6, 10
9	TPM_PhysicalEnable	**PP**	-
10	TPM_PhysicalDisable	**PP**	-
11	TPM_PhysicalSetDeactivated	**PP**	-
12	TPM_SetTempDeactivated	**U, PP**	R: 6, 10, 15 W: 6, 10
13	TPM_SetOperatorAuth	**PP**	W: 15
Admin Ownership			
14	TPM_TakeOwnership	**CO**	R: 1, 6, 8, 10, 12, 14 W: 2, 5, 6, 10, 11, 12
15	TPM_OwnerClear	**CO**	R: 8, 10, 12 Z: 2, 3, 5, 8, 9, 10, 11, 12, 13, 14, 15
16	TPM_ForceClear	**PP**	Z: 2, 3, 5, 8, 9, 10, 11, 12, 13, 14, 15
17	TPM_DisableOwnerClear	**CO**	R: 8, 10, 12
18	TPM_DisableForceClear	**NA**	-
19	TSC_PhysicalPresence	**NA**	-
20	TSC_ResetEstablishmentBit	**NA**	-
Capability			
21	TPM_GetCapability	**NA**	O: 4 (SHA-256 of public key)
22	TPM_SetCapability	**CO**	R: 6, 8, 10, 12 W: 6, 10
23	TPM_GetCapabilityOwner	**CO**	R: 6, 8, 10, 12
Administrative Functions & Management			

Services		Role	Keys and CSP access (R = read, W = write, O = output, Z = zeroize)
29	TPM_ResetLockValue	**CO**	R: 3, 6, 8, 10, 12 W: 6, 10
Storage			
30	TPM_Seal	**U**	R: 3, 6, 8, 10, 11 W: 6, 10
31	TPM_Unseal	**U**	R: 3, 6, 8, 10, 11 W: 6, 10
32	TPM_UnBind	**U**	R: 3, 6, 8, 10 W: 6, 10
33	TPM_CreateWrapKey	**U**	R: 3, 6, 8, 10, 14 W: 6, 10 O: 3 (private part is encrypted)
34	TPM_LoadKey2	**U**	R: 6, 8, 10, 11 W: 3, 6, 10
35	TPM_GetPubKey	**U**	R: 3, 6, 8, 10 W: 6, 10
Migration			
37	TPM_CreateMigrationBlob	**U**	R: 3, 6, 8, 10, 11 W: 6, 10
38	TPM_ConvertMigrationBlob	**U**	R: 3, 6, 8, 10 W: 6, 10
39	TPM_AuthorizeMigrationKey	**CO**	R: 3, 6, 8, 10, 11, 12 W: 6, 10
40	TPM_MigrateKey	**U**	R: 3, 6, 8, 10 W: 6, 10
41	TPM_CMK_SetRestrictions	**CO**	R: 3, 6, 8, 10, 12 W: 6, 10
42	TPM_CMK_ApproveMA	**CO**	R: 6, 8, 10, 11, 12 W: 6, 10
43	TPM_CMK_CreateKey	**U**	R: 2, 3, 6, 8, 10, 11, 14 W: 3, 6, 10 O: 3 (private part is encrypted)
44	TPM_CMK_CreateTicket	**CO**	R: 2, 3, 6, 8, 10, 12 W: 6, 10
45	TPM_CMK_CreateBlob	**U**	R: 3, 6, 8, 10, 11 W: 6, 10
46	TPM_CMK_ConvertMigration	**U**	R: 3, 6, 8, 10, 12 W: 6, 10
Cryptographic Functions			
52	TPM_SHA1Start	**NA**	**-**
53	TPM_SHA1Update	**NA**	**-**
54	TPM_SHA1Complete	**NA**	**-**
55	TPM_SHA1CompleteExtend	**NA**	**-**

Services		Role	Keys and CSP access (R = read, W = write, O = output, Z = zeroize)
56	TPM_Sign	**U**	R: 3, 6, 8, 10 W: 6, 10
57	TPM_GetRandom	**NA**	R: 6, 10 W: 6, 10
58	TPM_StirRandom	**NA**	R: 6, 10 W: 6, 10
Endorsement Key Handling			
64	TPM_ReadPubek	**NA**	**-**
65	TPM_OwnerReadInternalPub	**CO**	R: 1, 2, 6, 8, 10, 12 W: 6, 10
Identity Creation and Activation			
66	TPM_MakeIdentity	**CO**	R: 2, 6, 8, 10, 11, 12, 14 W: 6, 10 O: 3 (identity key, private part is encrypted)
67	TPM_ActivateIdentity	**CO**	R: 1, 6, 8, 10, 12 W: 6, 10
Integrity Collection and reporting			
68	TPM_Extend	**NA**	**-**
69	TPM_PCRRead	**NA**	**-**
71	TPM_PCR_Reset	**NA**	**-**
Changing Auth Data			
73	TPM_ChangeAuth	**U**	R: 6, 8, 10 W: 3, 6, 9, 10
74	TPM_ChangeAuthOwner	**CO**	R: 6, 8, 10, 12 W: 2, 6, 10, 12
Authorization sessions			
75	TPM_OIAP	**NA**	R: 6, 10 W: 6, 8, 10
76	TPM_OSAP	**NA**	R: 2, 3, 6, 9, 10, 12, 13 W: 6, 8, 10
77	TPM_DSAP	**NA**	R: 3, 6, 10, 11 W: 6, 8, 10
78	TPM_SetOwnerPointer	**NA**	**-**
Delegation			
79	TPM_Delegate_Manage	**CO**	R: 6, 8, 10, 12 W: 6, 8, 10
80	TPM_Delegate_CreateKeyDelegation	**U**	R: 3, 5, 6, 8, 10, 11, 12 W: 6, 10
81	TPM_Delegate_CreateOwnerDelegation	**CO**	R: 5, 6, 8, 10, 11, 12 W: 6, 10
82	TPM_Delegate_LoadOwnerDelegation	**CO**	R: 5, 6, 8, 10, 11, 12 W: 6, 10
83	TPM_Delegate_ReadTable	**NA**	**-**

Services		Role	Keys and CSP access (R = read, W = write, O = output, Z = zeroize)
84	TPM_Delegate_UpdateVerification	**CO**	R: 6, 8, 10, 11, 12 W: 6, 10
85	TPM_Delegate_VerifyDelegation	**NA**	R: 5, 6, 11
Non-Volatile Storage			
86	TPM_NV_DefineSpace	**CO**	R: 6, 8, 10, 12 W: 6, 9, 10 Z: 9 (if index previously defined and size = 0)
87	TPM_NV_WriteValue	**CO**	R: 6, 8, 10, 12 W: 6, 10
88	TPM_NV_WriteValueAuth	**U**	R: 6, 8, 9, 10 W: 6, 10
89	TPM_NV_ReadValue	**CO**	R: 6, 8, 10, 12 W: 6, 10
90	TPM_NV_ReadValueAuth	**U**	R: 6, 8, 9, 10 W: 6, 10
Session Management			
91	TPM_KeyControlOwner	**CO**	R: 6, 8, 10, 12 W: 3, 10
92	TPM_SaveContext	**NA**	R: 3, 5, 9, 11 Z: 8
93	TPM_LoadContext	**NA**	R: 3, 5, 9, 11
Eviction			
94	TPM_FlushSpecific	**NA**	Z: 3, 8
Timing Ticks			
95	TPM_GetTicks	**NA**	-
Transport Sessions			
97	TPM_EstablishTransport	**U**	R: 3, 6, 8, 10 W: 6, 10
98	TPM_ExecuteTransport	**U**	R: 6, 8, 10 W: 6, 10
Monotonic Counter			
100	TPM_CreateCounter	**CO**	R: 6, 8, 10, 12 W: 6, 10, 13
101	TPM_IncrementCounter	**U**	R: 6, 8, 10, 13 W: 6, 10
102	TPM_ReadCounter	**NA**	-
103	TPM_ReleaseCounter	**U**	R: 6, 8, 10, 13 W: 6, 10 Z: 8, 13
104	TPM_ReleaseCounterOwner	**CO**	R: 6, 8, 10, 12 W: 6, 10 Z: 8, 13
Signal Commands			

Services		Role	Keys and CSP access (R = read, W = write, O = output, Z = zeroize)
124	TPM_HASH_START	**NA**	**-**
125	TPM_HASH_DATA	**NA**	**-**
126	TPM_HASH_END	**NA**	**-**
Proprietary commands			
127	TPM_FieldUpgradeStart	**CO, PP**	R: 4, 6, 8, 10, 12 W: 6, 10
128	TPM_FieldUpgradeData (Uses service 144)	**NA**	**-**
129	TPM_SHA256Start	**NA**	**-**
130	TPM_SHA256Update	**NA**	**-**
131	TPM_SHA256Complete	**NA**	**-**
133	TPM_SetMode	**CO**	R: 6, 8, 10, 12 W: 6, 10 Z: 3, 14
Deprecated commands			
134	TPM_EvictKey	**NA**	Z: 3
135	TPM_Terminate_Handle	**NA**	-
136	TPM_DirWriteAuth	**CO**	R: 6, 8, 10, 11, 12 W: 9, 10
137	TPM_DirRead	**NA**	R: 9
138	TPM_ChangeAuthAsymStart	**U**	R: 3, 6, 8, 10 W: 6, 10
139	TPM_ChangeAuthAsymFinish	**U**	R: 6, 8, 10 W: 3, 6, 10
140	TPM_Reset	**NA**	**-**
141	TPM_OwnerReadPubek	**CO**	R: 1, 6, 8, 10, 12 W: 6, 10
142	TPM_DisablePubekRead	**CO**	R: 6, 8, 10, 12 W: 6, 10
143	TPM_LoadKey	**U**	R: 3, 6, 8, 10, 11 W: 6, 10
Non FIPS service			
144	Field upgrade de-obfuscation[1]	**NA**	**-**

[1] This service is not callable from TPM interface but is only used internally by TPM_FieldUpgradeData command. It consists in de-obfuscating data received by the TPM_FieldUpgradeData command with a non-FIPS approved algorithm.

3.3 Key management

3.3.1 *Key entry and output*

Next table indicates the approved method used to encrypt all secret and private keys (indicated by S for secret value and P for private key in type column), entered into or output from the cryptographic module.

Table 11 : Encrypted methods for secret and private keys input

Service	Parameter name	Type	Input or output	Encryption algorithm
TPM_LoadKey	inKey (private part)	P	Input	RSA-OAEP SHA1
TPM_LoadKey2	inKey (private part)	P	Input	RSA-OAEP SHA1
TPM_TakeOwnership	encOwnerAuth	S	Input	RSA-OAEP SHA1
	encSrkAuth	S	Input	RSA-OAEP SHA1
TPM_Seal	encAuth	S	Input	AES CTR 128
TPM_CreateWrapKey	dataUsageAuth	S	Input	AES CTR 128
	dataMigrationAuth	S	Input	AES CTR 128
	wrappedKey	P	Output	RSA-OAEP SHA1
TPM_CMK_CreateKey	dataUsageAuth	S	Input	AES CTR 128
	wrappedKey (private part)	P	Output	RSA-OAEP SHA1
TPM_EstablishTransport	secret	S	Input	RSA-OAEP SHA1
TPM_MakeIdentity	identityAuth	S	Input	AES CTR 128
TPM_Delegate_CreateKeyDelegation	delAuth	S	Input	AES CTR 128
TPM_Delegate_CreateOwnerDelegation	delAuth	S	Input	AES CTR 128
TPM_NV_DefineSpace	encAuth	S	Input	AES CTR 128
TPM_CreateCounter	encAuth	S	Input	AES CTR 128
TPM_SaveContext	contextBlob	P	Output	AES CTR 128
TPM_LoadContext	contextBlob	P	Input	AES CTR 128
TPM_CreateMigrationBlob	outData	P	Output	RSA-OAEP SHA1
TPM_ConvertMigrationBlob	inData	P	Input	RSA-OAEP SHA1
TPM_MigrateKey	inData	P	Input	RSA-OAEP SHA1
	outData	P	Output	RSA-OAEP SHA1
TPM_CMK_ConvertMigration	outData	P	Output	RSA-OAEP SHA1
TPM_ChangeAuth	encData	S	Input	AES CTR 128

3.3.2 *Key transport*

As indicated in the above table, the TPM supports two different algorithms for key transport. Relative security strength of each cryptographic algorithm supported by the module is indicated in the table below:

Table 12: Cryptographic Functions

Algorithm	Comparable number of bits of security
RSA-2048	112
AES-128[1]	128

RSA-2048 and AES-128 are used to transport RSA-2048 keys (security strength of the transport method is then greater or equal than the security strength of the keys transported).

AES-128 in CTR mode is also used in ADIP protocol to encrypt 160-bits authorization data.

RSA is used with OAEP SHA-1 padding scheme method to encrypt (wrap) and decrypt (unwrap) secrets and private keys, as indicated in Table 11 : Encrypted methods for secret and private keys input, with a parent key already loaded into the TPM.

AES is used in CTR mode to encrypt/decrypt with shared secret from OSAP session as key for all commands listed in Table 11 : Encrypted methods for secret and private keys input except for TPM_SaveContext and TPM_LoadContext that uses contextKey.

[1] AES is used in conjunction with HMAC-SHA-1 approved authentication method (scheme is compliant with **[SP800-38F]**)

4 SELF-TESTS

Self-tests run by the cryptographic module are split in two categories:

- Power-up self-tests
- Conditional self-tests

The power-on self-tests do not require operator intervention in order to run. Power-on self-tests execution always completes the full suite of self-tests in its entirety. Input activity is ignored and output activity is inhibited until self-tests have successfully completed.

The security module outputs an “error” Return Code via the status interface when the error state is entered due to a failed self-test. While in error state, security module does not perform any cryptographic functions and all data output via the data output interface are inhibited.

If power-on self-tests have passed successfully, no status is indicated but commands that require self-tests to be completed can be successfully executed.

4.1 Power-up tests list

Table 13 : Cryptographic algorithm KATs

Algorithm tested	Test description
SHA1	SHA1 computation on known data (16 bytes) and comparison of output to the expected digest (20 bytes)
SHA256	SHA256 computation on known data (16 bytes) and comparison of output to the expected digest (32 bytes)
HMAC SHA1	HMAC-SHA1 computation on known data (16 bytes) / known key (16 bytes, same value as data) and comparison of output to the expected MAC (20 bytes)
KDF SP800-108	KDFa (based on SHA1) computation on known data (16 bytes) / known label (“TEST”) and comparison of output to the expected value (32 bytes).
Hash DRBG	Hash DRBG is self-tested acordingly to **[SP800-90A]** §11.3**.** KAT is conducted on Instantiate, Reseed and Generate API in a single test sequence. A known seed value is used to instantiate the DRBG. Output of HDRBG (55 bytes) is compared to a reference value.
AES	AES CFB encryption is done on known data (32 bytes) / known key (16 bytes) and known IV (16 bytes, same value as key). The 32 bytes output data are compared to the expected reference data. If comparison succeeds, AES CFB decryption is done on encrypted data with same key & same IV as encryption. 32 bytes output are compared to the initial plaintext data.
RSA	A known key is loaded (2048 bits length). Signature RSASSA-PKCS1-v1_5 is generated on known data (20 bytes). Output of signature is compared to a reference signature. If comparison is successful, signature verification is performed. Failure state is entered if one of the step (generation or verification) fails.

Table 14 : TPM integrity tests

Algorithm tested	Test description
FW integrity	FW integrity is verified by computing an EDC (CRC-16 ISO 13239) and comparing it to reference values.
HW integrity	HW integrity is guaranteed via check of HW sensors.

4.2 Conditional tests list

Table 15 : TPM conditional tests

Algorithm tested	Test description
Hash-DRBG	Each 32 bytes of generated data are compared to the previous generated data. If data are equal, status is set to FAIL and error is returned.
NDRNG	TPM performs AIS31 statistical test verification on NDRNG output and continuous HW self-tests (AS09.42) on NDRNG 48-bits output sequence. If test fails, status is set to FAIL and error is returned.
FW load	During field upgrade procedure, several checks are performed before authorizing the FW to be upgraded: - Verification of signature (RSASSA-PSS) on the first data blob to ensure authentication of the FW - Verification of digest (SHA256) on each subsequent blob to guarantee integrity of the full FW.
RSA **key generation**	A new RSA key is generated or retrieved from pre-computed keys (done in BKG). Depending on the key purpose (signing or encrypting) indicated in TPM_KEY_USAGE structure, en/decryption or signing/verification is done on known data (16 bytes).

4.3 Verification

Successful completion of self-tests can be verified through use of TPM_GetTestResult command. If the first 4 bytes of response are equal to 0, self-tests completed successfully.

5 PHYSICAL SECURITY POLICY

The security module meets Physical Security protection requirements for FIPS level 1. Physical security at level 1 assumes no physical protection of CSPs. No action is required by the operator(s) to ensure that physical security is maintained. Some physical security protection mechanisms beyond the requirements for level 1 have been implemented and are described in “Mitigations of other attacks”.

Normal operating ranges are defined in the respective module datasheet **[ST33TPHF2ESPI DS]**:

- **Temperature:**

  The normal operating temperature range of the security module is defined in §1.2.

- **Voltage:**

  The normal operating voltage range of the security module is 1.8V or 3.3V (±10%).

- **Frequency:**

  The internal system clock is created by an internal oscillator.

Operation outside these ranges is not guaranteed, but physical security mechanisms are implemented to assure that CSPs remain protected from unauthorized disclosure, usage, modification or deletion.

6 OPERATIONAL ENVIRONMENT

Module operational environment is “limited modifiable” because TPM FW can only be modified through field upgrade service (use of TPM_FieldUpgradeStart and TPM_FieldUpgradeData commands). The Non-upgradable code blocks are non-modifiable.

FIPS 140-2 level 1 operational environment requirements of **[FIPS140-2]** section 4.6.1 are then not applicable to the security module.

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 require a separate FIPS 140-2 validation.

7 MITIGATIONS OF OTHER ATTACKS

The security module meets Physical Security protection requirements for FIPS level 1. Physical security at level 1 assumes no physical protection of CSPs. Physical security protection mechanisms beyond the level 1 requirements have been implemented and are described in this section.

7.1 Internal Tamper Detection

The security module contains an active metal shield that covers the internal TPM circuitry and memory components. Cutting, removing or modifying the shield layer will cause the TPM to Reset and enter a SHUTDOWN mode.

7.2 Environmental protection

The security module contains circuitry which will detect environmental conditions outside the range described in the product datasheet. Power supply voltage is continuously monitored. If conditions exist outside the range determined by the TPM tamper detection circuitry, the chip will reset and will enter a FAILURE mode. The chip will remain Reset and in FAIL mode as long as the environmental condition causing the tamper event persists.

8 REFERENCES

Reference	Document
[ST33TPHF2ESPI DS]	ST33TPHF2ESPI Datasheet, STMicroelectronics, December 2015
[TPM2E SCY]	ST33TPH2ESPI, Security guidelines for TPM configuration (1.3), STMicroelectronics, December 2015
[TPM Part1 r116]	TPM Main, Part 1, Design principles, Version 1.2 Level 2, rev 116, TCG
[TPM Part2 r116]	TPM Main, Part 2, TPM Structures, Version 1.2 Level 2, revision 116, TCG
[TPM Part3 r116]	TPM Main, Part 3, Commands, Version 1.2 Level 2, revision 116, TCG
[TIS 1.30]	TCG PC Client Specific TPM Interface Specification (TIS) – Version 1.3
[TPM 1.2 PPI]	Trusted Computing Group Physical Presence Interface Specification; Specification, version 1.2; Version 1.20; Revision 1.00; February 10, 2011
[FIPS140-2]	FIPS PUB 140-2, *Security Requirements for Cryptographic Modules* / National Institute of Standards and Technology (NIST), CHANGE NOTICES (12-03-2002)
[FIPS DTR]	National Institute of Standards and Technology and Communications Security, *Derived Test Requirements(DTR) for FIPS PUB 140-2, Security Requirements for Cryptographic Modules*
[FIPS IG]	National Institute of Standards and Technology and Communications Security, *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*
[FIPS 180-4]	National Institute of Standards and Technology, *Secure Hash Standard*, Federal Information Processing Standards Publication 180-4, March 2012
[FIPS 186-4]	National Institute of Standards and Technology, *Digital Signature Standard (DSS),* Federal Information Processing Standards Publication 186-4, July 2013
[FIPS 197]	National Institute of Standards and Technology, *Advanced Encryption Standard (AES),* Federal Information Processing Standards Publication 197, November 2001
[SP800-135]	National Institute of Standards and Technology, *Existing Application-Specific Key Derivation Function Validation System*, September 2015.
[SP800-108]	National Institute of Standards and Technology, *Recommendation for Key Derivation Using Pseudorandom Functions*, October 2009.

Reference	Document
[SP800-131A]	National Institute of Standards and Technology, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, 11/06/15.
[SP198-1]	National Institute of Standards and Technology, *The Keyed-Hash Message Authentication Code,* NIST Computer Security Division Page 3 07/26/2011, *(HMAC)*, Federal Information Processing Standards Publication 198-1, July, 2008
[SP800-90A]	National Institute of Standards and Technology, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, January 2012.
[SP800-38F]	National Institute of Standards and Technology, *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping,* December 2012.

9 ACRONYMS

Term	Definition
ADIP	Authorization-Data Insertion Protocol
AES	Advanced Encryption Standard
CO	Crypto Officer
DES	Data Encryption Standard
DSAP	Delegate Specific Authorization Protocol
EK	Endorsement Key
FIPS	Federal Information Processing Standard
FUM	Field Upgrade Mode
GPIO	General Purpose I/O
HMAC	Keyed-Hashing for Message Authentication
NIST	National Institute of Standards and Technology
NV	Non-volatile (memory)
OIAP	Object-Independent Authorization Protocol
OSAP	Object Specific Authorization Protocol
PCR	Platform Configuration Register
RSA	Rivest Shamir Adelman
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
SHA	Secure Hash Algorithm
SPI	Serial Peripheral Interface
SRK	Storage Root Key
TCG	Trusted Computed Group
TPM	Trusted Platform Module
TSS	TPM Software Stack

WARNING AND DISCLAIMER

CONFIDENTIALITY OBLIGATIONS:

THIS DOCUMENT IS A NON-PROPRIETARY SECURITY POLICY AND CAN BE FREELY DISTRIBUTED

Please Read Carefully:

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

www.st.com