STMicroelectronics

Trusted Platform Module ST33KTPM2X / ST33KTPM2XSPI

FIPS 140-3 Non-Proprietary Security Policy

Document Version: 01-01
Date: 2025-12-01

Table of Contents

1.1 Overview ... 5

1.2 Security Levels ... 5

2 Cryptographic Module Specification ... 6

2.1 Description ... 6

2.2 Tested and Vendor Affirmed Module Version and Identification ... 8

2.3 Excluded Components ... 11

2.4 Modes of Operation ... 11

2.5 Algorithms ... 12

2.6 Security Function Implementations ... 14

2.7 Algorithm Specific Information ... 19

2.8 RBG and Entropy ... 19

2.9 Key Generation ... 20

2.10 Key Establishment ... 20

2.11 Industry Protocols ... 20

3 Cryptographic Module Interfaces ... 21

3.1 Ports and Interfaces ... 21

3.2 Pinout description ... 22

4 Roles, Services, and Authentication ... 24

4.1 Authentication Methods ... 24

4.2 Roles ... 25

4.3 Approved Services ... 25

4.4 Non-Approved Services ... 69

4.5 External Software/Firmware Loaded ... 74

5 Software/Firmware Security ... 75

5.1 Integrity Techniques ... 75

5.2 Initiate on Demand ... 75

6 Operational Environment ... 76

6.1 Operational Environment Type and Requirements ... 76

7 Physical Security ... 77

7.1 Mechanisms and Actions Required ... 77

7.2 User Placed Tamper Seals ... 77

7.3 Filler Panels ... 77

7.4 Fault Induction Mitigation ... 77

7.5 EFP/EFT Information 78
7.6 Hardness Testing Temperature Ranges 78
8 Non-Invasive Security 79
8.1 Mitigation Techniques 79
9 Sensitive Security Parameters Management 80
9.1 Storage Areas 80
9.2 SSP Input-Output Methods 80
9.3 SSP Zeroization Methods 82
9.4 SSPs 82
9.5 Transitions 96
9.6 Additional Information 96
10 Self-Tests 97
10.1 Pre-Operational Self-Tests 97
10.2 Conditional Self-Tests 97
10.3 Periodic Self-Test Information 99
10.4 Error States 100
11 Life-Cycle Assurance 101
11.1 Installation, Initialization, and Startup Procedures 101
11.2 Administrator Guidance 102
11.3 Non-Administrator Guidance 102
11.4 Design and Rules 102
11.5 Maintenance Requirements 103
11.6 End of Life 103
12 Mitigation of Other Attacks 104
References and Definitions 105

List of Tables

Table 1: Security Levels 5
Table 2: Tested Module Identification – Hardware 8
Table 3 – KE2 Module Configuration 9
Table 4 – KE3 Module Configuration 9
Table 5 – KG8 Module Configuration 9
Table 6 – KG9 Module Configuration 10
Table 7 – KJ5 Module Configuration 10
Table 8 – KJ0 Module Configuration 10
Table 9 – KJ1 Module Configuration 11
Table 10: Modes List and Description 12
Table 11: Approved Algorithms 13
Table 12: Vendor-Affirmed Algorithms 13
Table 13: Non-Approved, Allowed Algorithms with No Security Claimed 13
Table 14: Non-Approved, Not Allowed Algorithms 14
Table 15: Security Function Implementations 19
Table 16: Entropy Certificates 19
Table 17: Entropy Sources 20
Table 18: Ports and Interfaces 21
Table 19 – UFQFPN32 Pins Definition 23
Table 20: Authentication Methods 24
Table 21: Roles 25
Table 22 – Mapping between services 25
Table 23: Approved Services 67
Table 24: Non-Approved Services 73
Table 25: Mechanisms and Actions Required 77
Table 26: EFP/EFT Information 78
Table 27: Hardness Testing Temperatures 78
Table 28: Storage Areas 80
Table 29: SSP Input-Output Methods 81
Table 30: SSP Zeroization Methods 82
Table 31: SSP Table 1 84
Table 32: SSP Table 2 88
Table 33 – Security Strength of a Key Depending on the Underlying Algorithm Used and its Size 96
Table 34: Pre-Operational Self-Tests 97
Table 35: Conditional Self-Tests 98
Table 36: Pre-Operational Periodic Information 99
Table 37: Conditional Periodic Information 100
Table 38: Error States 100
Table 39 – List of policy commands to use in a policy session 102
Table 40 – References 106
Table 41 – Acronyms and Definitions 107

List of Figures

Figure 1 – HW block diagram 7
Figure 2 – UFQFPN32 Package 8
Figure 3 – UFQFPN32 Pinout Diagram 23
Figure 4 – Firmware block diagram.. 75

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for the STMicroelectronics Trusted Platform Module ST33KTPM2X / ST33KTPM2XSPI. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 2 module.

1.2 Security Levels

The FIPS 140-3 security levels for the module are listed in table below:

Section	Title	Security Level
1	General	2
2	Cryptographic module specification	2
3	Cryptographic module interfaces	2
4	Roles, services, and authentication	2
5	Software/Firmware security	2
6	Operational environment	N/A
7	Physical security	3
8	Non-invasive security	N/A
9	Sensitive security parameter management	2
10	Self-tests	2
11	Life-cycle assurance	2
12	Mitigation of other attacks	N/A
	Overall Level	2

Table 1: Security Levels

2 Cryptographic Module Specification

The ST33KTPM2X / ST33KTPM2XSPI module, hereafter denoted as the Module, is a fully integrated security module implementing the revision 1.59 of the Trusted Computing Group (TCG) specification for Trusted Platform Modules (TPM) version 2.0.

2.1 Description

Purpose and Use:

The Module is intended for use by US Federal agencies or other markets that require FIPS 140-3 validated level 2. The Module is designed to be integrated into personal computers or any other embedded electronic systems. TPM is primarily used for cryptographic keys generation, keys storage, keys management and secure storage for digital certificates.

Module Type: Hardware

Module Embodiment: SingleChip

Module Characteristics:

Cryptographic Boundary:

The cryptographic boundary of the Module is defined as the perimeter of the IC package and is represented in the next figure. The Module is composed of:

- Two CPU cores, each including an MPU.
- Memories (RAMs, Flash and ROM) that store data or FW.
- HW accelerators for CRC (16 and 32-bits), symmetric cryptographic operations (AES) and asymmetric cryptographic operations (RSA/ECC).
- A clock generator and timers.
- An entropy source covered by an ESV Certificate.
- SPI and I^{2}C master/slave blocks.
- An administration block dedicated to chip security configuration and alarms detection.

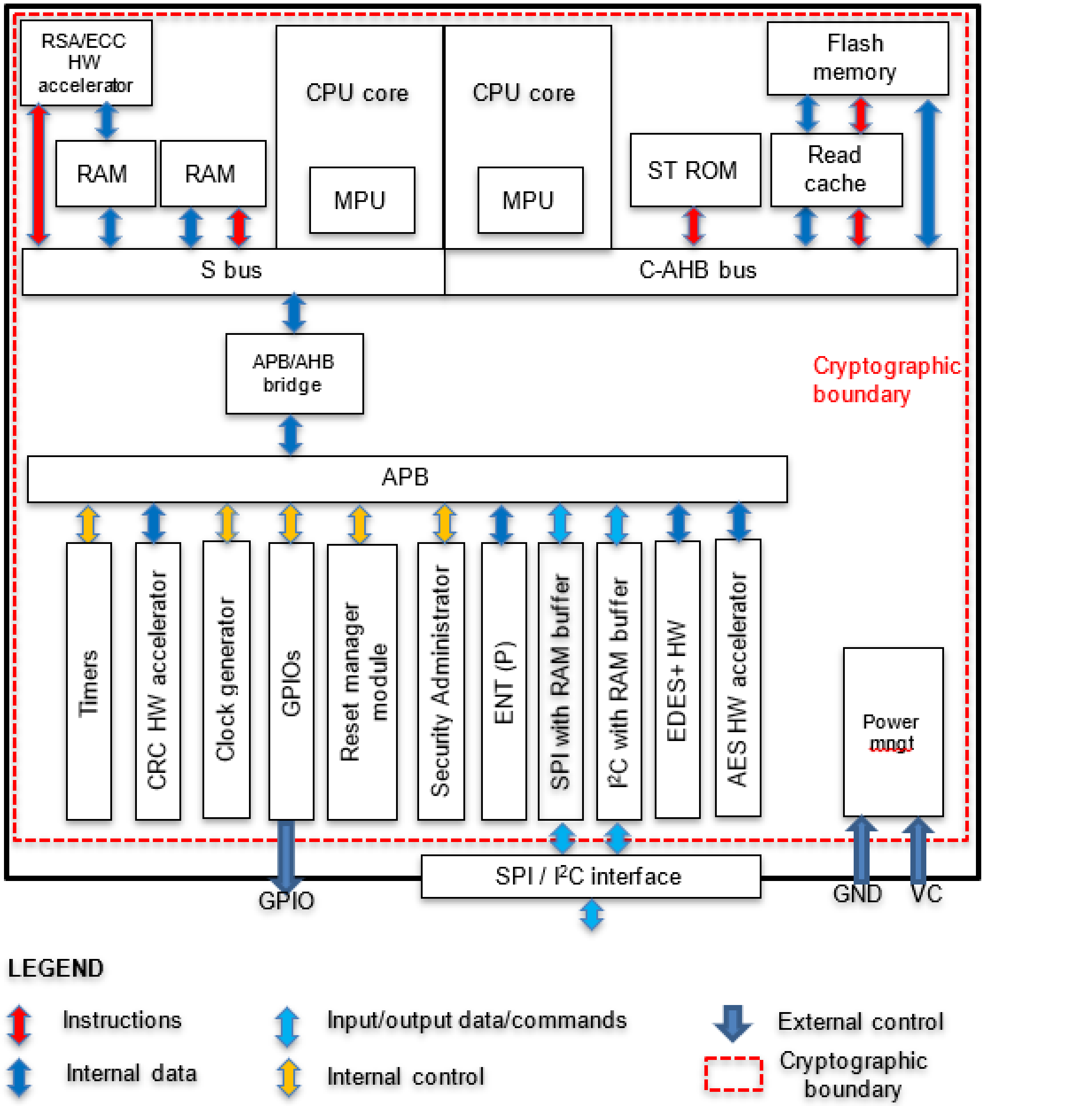

Figure 1 – HW block diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

The operating environments covered by the FIPS 140-3 validation are summarized in the table below:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
ST33KTPM2X	ST33K1M5T revC & revD	9.512 (dec.) 0x00.09.02.00 (hex.)	ST33K1M5T	SPI or I2C The interface is exclusive and selectable dynamically during product boot.
ST33KTPM2XSPI	ST33K1M5T revC & revD	9.512 (dec.) 0x00.09.02.00 (hex.)	ST33K1M5T	SPI

Table 2: Tested Module Identification – Hardware

ST33KTPM2X and ST33KTPM2XSPI are manufactured in the UFQFPN32 package:

- UFQFPN32
  - Ultra-thin pitch Quad Flat No-lead 32-pin
  - 5 x 5 mm

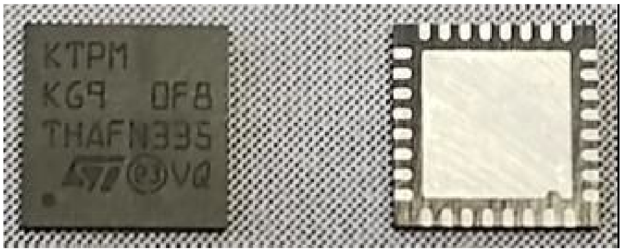

Figure 2 – UFQFPN32 Package

Products list:

The Module configurations indicated in Section 2.2 are defined into several manufactured products listed hereafter.

The default firmware version of KE2 is 9.256. To operate with firmware version 9.512, module must be first field upgraded to 9.512.

	Module Configuration*

Module name / HW P/N	ST33KTPM2XSPI
Package	UFQFPN32
Interface	SPI
Marking	KTPM KE2
FW version	00.09.02.00 (9.512)
TPM2.0 revision	1.59

Table 3 – KE2 Module Configuration

The default firmware version of KE3 is 9.256. To operate with firmware version 9.512, module must be first field upgraded to 9.512.

	Module Configuration*
Module name / HW P/N	ST33KTPM2X
Package	UFQFPN32
Interface	SPI / I^{2}C
Marking	KTPM KE3
FW version	00.09.02.00 (9.512)
TPM2.0 revision	1.59

Table 4 – KE3 Module Configuration

The default firmware version of KG8 is 9.257. To operate with firmware version 9.512, module must be first field upgraded to 9.512.

	Module Configuration*
Module name / HW P/N	ST33KTPM2XSPI
Package	UFQFPN32
Interface	SPI
Marking	KTPM KG8
FW version	00.09.02.00 (9.512)
TPM2.0 revision	1.59

Table 5 – KG8 Module Configuration

The default firmware version of KG9 is 9.257. To operate with firmware version 9.512, module must be first field upgraded to 9.512.

	Module Configuration*

Module name / HW P/N	ST33KTPM2X
Package	UFQFPN32
Interface	SPI / I^{2}C
Marking	KTPM KG9
FW version	00.09.02.00 (9.512)
TPM2.0 revision	1.59

Table 6 – KG9 Module Configuration

The default firmware version of KJ5 is 9.258. To operate with firmware version 9.512, module must be first field upgraded to 9.512.

	Module Configuration*
Module name / HW P/N	ST33KTPM2XSPI
Package	UFQFPN32
Interface	SPI
Marking	KTPM KJ5
FW version	00.09.02.00 (9.512)
TPM2.0 revision	1.59

Table 7 – KJ5 Module Configuration

	Module Configuration*
Module name / HW P/N	ST33KTPM2XSPI
Package	UFQFPN32
Interface	SPI
Marking	KTPM KJ0
FW version	00.09.02.00 (9.512)
TPM2.0 revision	1.59

Table 8 – KJ0 Module Configuration

	Module Configuration*
Module name / HW P/N	ST33KTPM2X

Package	UFQFPN32
Interface	SPI / I^2C
Marking	KTPM KJ1
FW version	00.09.02.00 (9.512)
TPM2.0 revision	1.59

Table 9 – KJ1 Module Configuration

The current FIPS 140-3 level 2 Security Policy applies to the Module configurations listed above when the Module is configured in FIPS 140-3 level 2 mode with the command TPM2_SetCapability. For the configurations supporting both SPI and I^2C interfaces, the selection of the mode is done during the boot of the Module.

2.3 Excluded Components

N/A

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Normal mode	TPM is in normal operation mode when all pre-operational and conditional self-tests (apart from FW load and PCT tests) are complete. All approved services are usable. The corresponding indicator reports if the service uses an approved cryptographic algorithm or security function.	Approved	TPM2_GetCapability (capability = TPM_CAP_VENDOR_PROPERTIES) with the sub-capability TPM_SUBCAP_VENDOR_TPMA_MODES = 0x7 shall be used. It outputs a 2-bit indicator equals to 01b if the module is in an approved mode of operation
Non-approved mode of operation	The module enters a non-approved mode if one of the non-approved	Non-Approved	TPM2_GetCapability (capability = TPM_CAP_VENDOR_PROPERTIES) with the sub-capability TPM_SUBCAP_VENDOR_TPMA_MODES = 0x7 shall be used. It outputs a 2-bit indicator

Mode Name	Description	Type	Status Indicator
	services is used by the operator.		equals to 10b if the module is in a non-approved mode of operation

Table 10: Modes List and Description

2.5 Algorithms

Approved Algorithms:

The Module implements the Approved cryptographic algorithms listed in the table below.

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5356	-	SP 800-38A
AES-CFB128	A5356	-	SP 800-38A
AES-CTR	A5356	-	SP 800-38A
AES-ECB	A5356	-	SP 800-38A
AES-OFB	A5356	-	SP 800-38A
ECDSA KeyGen (FIPS186-4)	A5358	-	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A5358	-	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A5358	-	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A5358	-	FIPS 186-4
Hash DRBG	A5351	-	SP 800-90A Rev. 1
HMAC-SHA-1	A5355	-	FIPS 198-1
HMAC-SHA2-256	A5355	-	FIPS 198-1
HMAC-SHA2-384	A5355	-	FIPS 198-1
HMAC-SHA2-512	A5355	-	FIPS 198-1
HMAC-SHA3-256	A5355	-	FIPS 198-1
HMAC-SHA3-384	A5355	-	FIPS 198-1
KAS-ECC Sp800-56Ar3	A5358	-	SP 800-56A Rev. 3
KDF SP800-108	A5354	-	SP 800-108 Rev. 1
KTS-IFC	A5357	-	SP 800-56B Rev. 2
LMS SigVer	A5360	-	SP 800-208
RSA Decryption Primitive Sp800-56Br2 (CVL)	A5357	-	SP 800-56B Rev. 2
RSA KeyGen (FIPS186-5)	A5357	-	FIPS 186-5
RSA SigGen (FIPS186-5)	A5357	-	FIPS 186-5
RSA SigVer (FIPS186-5)	A5357	-	FIPS 186-5
SHA-1	A5352	-	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA-1	A5353	-	FIPS 180-4
SHA2-256	A5352	-	FIPS 180-4
SHA2-256	A5353	-	FIPS 180-4
SHA2-384	A5352	-	FIPS 180-4
SHA2-384	A5353	-	FIPS 180-4
SHA2-512	A5352	-	FIPS 180-4
SHA2-512	A5353	-	FIPS 180-4
SHA3-256	A5352	-	FIPS 202
SHA3-384	A5352	-	FIPS 202

Table 11: Approved Algorithms

Vendor-Affirmed Algorithms:

The Module implements the Vendor Affirmed cryptographic algorithms listed.

Name	Properties	Implementation	Reference
CKG	Key Type:Symmetric	N/A	Section 4, Example 1 of [133r2]; IG D.H
CKG-Asym	Key Type:Asymmetric	N/A	Section 4, Example 1 of [133r2]; IG D.H

Table 12: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

The Module implements the Non-Approved, Allowed cryptographic Algorithms with No Security Claimed.

Name	Caveat	Use and Function
XOR	No security claimed per IG 2.4.A with the example of scenario #1. The algorithm: * is not used except for this purpose * does not access or share CSPs in a way that counters the requirements of the IG * not intended to be used as a security function. * can't be confused for a security function	Obfuscation of input or output data

Table 13: Non-Approved, Allowed Algorithms with No Security Claimed

Non-Approved, Not Allowed Algorithms:

The Module implements the Non-Approved, Not Allowed cryptographic algorithms listed.

Name	Use and Function
ECC BN P-256 (non-compliant)	Key generation, digital signature generation based on ECC BN P-256
ECC derived keys (non-compliant)	Secret exchange or digital signature generation/verification
ECDAA (non-compliant)	Key generation, digital signature generation
ECDSA (non-compliant)	Digital signature with an ECC signing key generated with an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme = TPM_ALG_NULL), derived from a derivation parent key, or a key loaded in the NULL hierarchy
ECSchnorr (non-compliant)	Key generation, digital signature generation and verification
HMAC (non-compliant)	Key length < 112 bits for message authentication
KAS (non-compliant)	Key agreement with an ECC key that has an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme = TPM_ALG_NULL)
KBKDF (non-compliant)	Non-Approved key derivation usage
KTS-IFC (non-compliant)	Key encapsulation with an RSA decryption key that has an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme = TPM_ALG_NULL)
RSA (non-compliant)	1024-bit RSA digital signature generation or with a key loaded in the Null hierarchy
RSA with no padding mode (null scheme) (non-compliant)	Key transport
RSAES-PKCS1-v1_5 (non-compliant)	Key transport
SHA-1 (non-compliant)	Digital signature generation

Table 14: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Next table shows the Security Function Implementations that the Module implements:

Name	Type	Description	Properties	Algorithms
KeyGen	AsymKeyPair-KeyGen	Key-Pair Generation	Publications:FIPS 186-5	ECDSA KeyGen (FIPS186-4): (A5358) CKG-Asym: () Key Type: Asymmetric RSA KeyGen (FIPS186-5): (A5357)
KeyVer	AsymKeyPair-KeyVer	Key-Pair Verification	Publications:FIPS 186-5	ECDSA KeyVer (FIPS186-4): (A5358)
KeyVal	AsymKeyPair-PubKeyVal	Key-pair Validation	Publications:186-5	KAS-ECC Sp800-56Ar3: (A5358) Function: Full Validation KTS-IFC: (A5357) Function: partialVal
AES-ENC	BC-UnAuth	Unauthenticated Encryption	Publication:FIPS 197	AES-CBC: (A5356) AES-CFB128: (A5356) AES-CTR: (A5356) AES-ECB: (A5356) AES-OFB: (A5356)
AES-DEC	BC-UnAuth	Unauthenticated Decryption	Publication:FIPS 197	AES-CBC: (A5356) AES-CFB128: (A5356) AES-CTR: (A5356) AES-ECB: (A5356) AES-OFB: (A5356)
SigGen	DigSig-SigGen	Signature Generation	Publication:FIPS 186-5	ECDSA SigGen

Name	Type	Description	Properties	Algorithms
				(FIPS186-4): (A5358) RSA SigGen (FIPS186-5): (A5357) SHA2-256: (A5352) SHA2-384: (A5352) SHA2-512: (A5352) SHA3-256: (A5352) SHA3-384: (A5352)
SigVer	DigSig-SigVer	Signature Verification	Publications:FIPS 186-5	LMS SigVer: (A5360) ECDSA SigVer (FIPS186-4): (A5358) RSA SigVer (FIPS186-5): (A5357) SHA2-256: (A5352) SHA2-384: (A5352) SHA2-512: (A5352) SHA3-256: (A5352) SHA3-384: (A5352)
DRBG	DRBG	Random Number Generation	Publication: :SP800-90A	Hash DRBG: (A5351) Method: SHA2-256 SHA2-256: (A5352)
ENT-ESV	ENT-ESV	ESV	Publications:SP800-90B	SHA2-256: (A5352) Conditioning Component: SHA2-256

Name	Type	Description	Properties	Algorithms
KAS	KAS-Full	Key establishment	Publications:SP 800-56A, Rev 3	KAS-ECC Sp800-56Ar3: (A5358) Schemes: fullUnified, onePassDH KDF: oneStepKDF SHA-1: (A5352) SHA2-256: (A5352) SHA2-384: (A5352) SHA2-512: (A5352) SHA3-256: (A5352) SHA3-384: (A5352)
KTS-IFC	KTS-Encap	Key Encapsulation	Publication:SP 800-56B rev 2, IG D.G Method:KTS-OAEP-basic	KTS-IFC: (A5357) RSA Decryption Primitive Sp800-56Br2: (A5357)
KTS	KTS-Wrap	Key transport	Publication:SP 800-38F, IG D.G	HMAC-SHA2-256: (A5355) AES-CFB128: (A5356)
KBKDF	KBKDF	Key-Based Key Derivation	Publications:SP800-108	KDF SP800-108: (A5354) SHA-1: (A5353) SHA2-256: (A5353) SHA2-384: (A5353) SHA2-512: (A5353) SHA3-256: (A5352) SHA3-384: (A5352)

Name	Type	Description	Properties	Algorithms
MAC	MAC	Message Authentication	Publication:FIPS198	HMAC-SHA-1: (A5355) HMAC-SHA2-256: (A5355) HMAC-SHA2-384: (A5355) HMAC-SHA2-512: (A5355) HMAC-SHA3-256: (A5355) HMAC-SHA3-384: (A5355) SHA-1: (A5352) SHA2-256: (A5352) SHA2-384: (A5352) SHA2-512: (A5352) SHA3-256: (A5352) SHA3-384: (A5352)
SHA	SHA	Secure Hash	Publications:FIPS 180-4, FIPS 202	SHA-1: (A5353, A5352) SHA2-256: (A5352, A5353) SHA2-384: (A5352, A5353) SHA2-512: (A5352, A5353) SHA3-256: (A5352) SHA3-384: (A5352)
CKG	CKG	Symmetric Key Generation	Publications:SP800-133rev2, Section 4; IG D.H	Hash DRBG: (A5351)

Name	Type	Description	Properties	Algorithms
KAS-KeyGen	KAS-KeyGen	KAS-ECC Key Generation	Publication:SP800-56Arev3	KAS-ECC Sp800-56Ar3: (A5358)

Table 15: Security Function Implementations

2.7 Algorithm Specific Information

Notes:

KAS [56Ar3] - Per [IG] D.F Scenario 2 path (2), compliant key agreement scheme where testing is performed end-to-end for the shared secret computation and a KDF compliant with <KDA>.
With/without key confirmation.

2.8 RBG and Entropy

The Module implements:

- A Hash-DRBG (#A5351) based on SHA2-256 and compliant with the [90A] standard (state is referred as drbgState in SSPs table). It is seeded at each module start-up with 512 bits generated from the internal, physical entropy source (ESV Cert. #E41). The entropy source provides 0.81926 bits of entropy per bit sampled, which results in a total of 419 bits of entropy (i.e., 512 * 0.81926) being used to instantiate the DRBG. This is sufficient to claim a full security strength of 256 bits.
- Hash-DRBG is used for any generation of random values used as SSP in a cryptographic operation. It can be reseeded by using the service TPM2_StirRandom.
- A transient Hash-DRBG based on SHA2-256 and compliant with the [90A] standard (state is referred as tdrbgState in SSPs table) involved only in primary keys generation and seeded as defined in [TPM2.0 Part1] and [TPM2.0 Part3].
- An entropy source as detailed below:

Cert Number	Vendor Name
E41	STMicroelectronics

Table 16: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Trusted Platform Module ST33KTPM2X, ST33KTPM2XSPI, ST33KTPM2XI2C,	Physical	ST33K1M5T/A platforms	1 bit	0.819266 bits	A5352 (SHA2-256)

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
ST33KTPM2A, ST33KTPM2I entropy source					

Table 17: Entropy Sources

2.9 Key Generation

For Key Generation methods, see Section 2.6 Security Function Implementations above.

2.10 Key Establishment

Key Agreement Information

For Key Establishment methods, see Section 2.6 Security Function Implementations above.

Key Transport Information

For Key Transport methods, see Section 2.6 Security Function Implementations above.

2.11 Industry Protocols

The Module does not implement any Industry Protocols.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

The Module’s ports and associated FIPS-defined logical interface categories are listed below.

Physical Port	Logical Interface(s)	Data That Passes
SPI_NSS / SPI_CLK / SPI_MOSI / I2C_SCL / I2C_SDA / RESET / PP	Control Input	Control parts of the TPM commands provided to the security module. It concerns all bytes of a command except plaintext data, ciphertext data and SSPs (entered with the data input interface).
SPI_NSS / SPI_CLK / SPI_MISO / I2C_SCL / I2C_SDA / PIRQ	Control Output	Control parts of the TPM responses output by the security module. It concerns all bytes of a response except plaintext data, ciphertext data and SSPs (output with the data output interface) and except the responseCode of a response (output with the status output interface)
SPI_NSS / SPI_CLK / SPI_MISO / I2C_SCL / I2C_SDA / PIRQ	Status Output	Status output by the security module (responseCode parameter of a response)
SPI_NSS / SPI_CLK / SPI_MOSI / I2C_SCL / I2C_SDA	Data Input	Data (plaintext data, ciphertext data and SSPs) provided to the security module as part of an input processing command
SPI_NSS / SPI_CLK / SPI_MISO / I2C_SCL / I2C_SDA	Data Output	Data (plaintext data, ciphertext data and SSPs) output by the security module as part of the response to a processing command
VCC / GND	Power	Power interface of the security module

Table 18: Ports and Interfaces

Additional details concerning the ports and interfaces of TPM:

1. Control and data inputs are multiplexed over the same physical interface. Control and data are distinguished by properly parsing input TPM command parameters according to input structures description, indicated for each command in [TPM2.0 Part3]. Some commands only deal with control input and status output parameters.

2. Status, data and control output are multiplexed over the same physical interface. Status, data and control are distinguished by properly setting output TPM response parameters according to output structures description, indicated for each command in [TPM2.0 Part3].
3. 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".
4. While performing key generation or key zeroisation (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.
5. To prevent the inadvertent output of CSPs in plaintext form on TPM2_Duplicate, the two following independent internal actions are performed:
    a. Verification of the encryptedDuplication attribute of the key to be duplicated
    b. Verification of the handle of the new parent of the key to be duplicated

    encryptedDuplication attribute must be set to 0 and new handle must be set to the null handle to authorize outputting the private part of the key in plaintext form.
6. 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.
7. The status output interface remains active during the error state to output the status of the security module with the service TPM2_GetCapability and TPM2_GetTestResult.

3.2 Pinout description

The pin layout for the UFQFPN32 package is shown in the next figure.

The ST33KTPM2X security module supports both SPI and I^{2}C physical interfaces but only one interface is configured during TPM boot. The interface configured remains active until the next module reset.

The ST33KTPM2XSPI security module only supports the SPI physical interface.

UFQFPN32 configuration

The pin layout for the UFWFPN32 package is shown in the next figure.

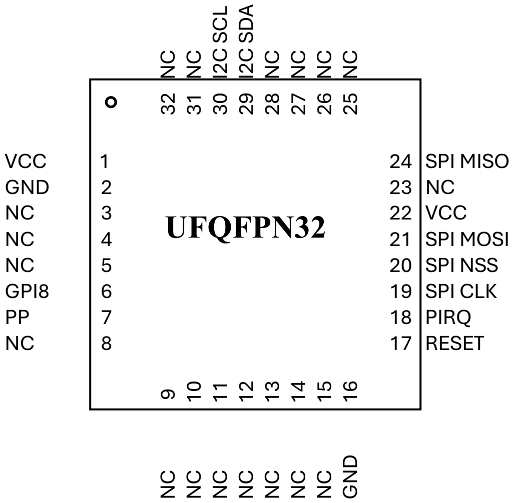

Figure 3 – UFQFPN32 Pinout Diagram

The table below gives a description of the products pins.

Signal	Type	Description*
VCC	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.
RESET	Input	Reset used to re-initialize the device
I2C SCL / GPIO5	Input or Input/Output	I^{2}C serial clock (Open drain with no weak pull-up resistor) or GPIO if SPI interface is selected
I2C SDA / GPIO6	Input/Output	I^{2}C serial data (Open drain with no weak pull-up resistor) or GPIO if SPI interface is selected
PIRQ	Output	IRQ used by TPM to generate an interrupt
SPI CLK / GPIO1	Input or Input/Output	SPI serial clock (output from master) or GPIO if I^{2}C interface is selected
SPI NSS / GPIO2	Input or Input/Output	SPI slave select (active low; output from master) or GPIO if I^{2}C interface is selected
SPI MISO / GPIO0	Output or Input/Output	SPI Master Input, Slave Output (output from slave) or GPIO if I^{2}C interface is selected
SPI MOSI / GPIO3	Input or Input/Output	SPI Master Output, Slave Input (output from master) or GPIO if I^{2}C interface is selected
GPI8	Input	GPI default to low. The level of this pin on the rising edge of the RESET signal is used to determine the physical interface to use (high level corresponds to SPI configuration and low-level to I^{2}C)
PP	Input	**Physical presence**, active high, internal pull-down. Used to indicate Physical Presence to the TPM.
NC	-	**Not Connected:** connected to the die but not usable. May be left unconnected. Internal pull-down.

Table 19 – UFQFPN32 Pins Definition

4 Roles, Services, and Authentication

4.1 Authentication Methods

The Module implements the following authentication techniques in accordance with the Level 2 requirements:

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Challenge-response authentication	The challenge-response mechanism uses an authorization value (authValue) as HMAC key or part of an HMAC key. The authValue is entered into the Module during the creation/loading of an object (key, NV index) or during replacement of the default value (hierarchies). The Module enforces a minimum size of 14 bytes.	MAC	Minimum strength is reached with an authValue of 14 bytes: 1/2^112 = 1.92*10^-34	Probability of a successful random attempt during a one-minute period is equal to 60000*1.92*10^-34 = 1.15*10^-29 (considering 60000 trials per minute). Assuming a minimum command duration of 1ms, 60000 trials can be executed during a one-minute period.
Enhanced authorization	Enhanced authorization includes a policy command (i.e., TPM2_PolicyAuthValue, TPM2_PolicySigned, TPM2_PolicyAuthorize, TPM2_PolicySecret, TPM2_PolicyTicket) requiring the knowledge of an authValue or the proof of the ownership of a signing key. It can also be a bound session, which also requires proving knowledge of an authValue of an object.	MAC or SigVer	Minimum strength is reached with an authValue of 14 bytes: 1/2^112 = 1.92*10^-34 or an RSA 2048 signature with a security strength of 112 bits	Probability of a successful random attempt during a one-minute period is equal to 60000*1.92*10^-34 = 1.15*10^-29 (considering 60000 trials per minute). Assuming a minimum command duration of 1ms, 60000 trials can be executed during a one-minute period.

Table 20: Authentication Methods

4.2 Roles

The Roles Table below lists all operator roles supported by the Module.

Name	Type	Operator Type	Authentication Methods
Crypto officer (CO)	Role	Administrator of the Module	Challenge-response authentication Enhanced authorization
User (U)	Role	User of the Module	Challenge-response authentication Enhanced authorization

Table 21: Roles

The Module does not provide a maintenance role or maintenance interface and does not support concurrent operators. The role is implicitly selected by the TPM operator on service execution by proving the knowledge of the enhanced authorization commands sequence and/or the authorization value of an object.

4.3 Approved Services

All services are accessible under the roles defined above and no specific access rights are considered to operate with keys and SSPs. Full services inputs and outputs are defined in [TPM2.0 Part3]. Next table indicates how mandatory services of [ISO/IEC 19790] (§7.4.3.1) are mapped to security module's services:

Mandatory service requested from [ISO/IEC 19790]*	Corresponding services from the security module
Show module's versioning information	TPM2_GetCapability
Show status	TPM2_GetTestResult
Perform self-tests	TPM2_SelfTest
Perform approved security functions	See approved services listed in next table
Perform zeroization	See services listed in section 9.3 SSP Zeroization Methods.

Table 22 – Mapping between services

All approved services implemented by the Module are listed in the table below:

The SSPs modes of access shown in the table below are defined as:

- G = Generate: The Module generates or derives the SSP.
- R = Read: The SSP is read from the Module (e.g., the SSP is output).
- W = Write: The SSP is updated, imported, or written to the Module (SSP is input).
- E = Execute: The Module uses the SSP in performing a cryptographic operation.
- Z = Zeroize: The Module zeroizes the SSP

Some details about information in the table:

- In “Name” column, **(I)** indicates that service is usable with sessions integrity mechanism, **(E)** indicates that service is usable with encryption session (encryption of 1st parameter of response), **(D)** indicates that service is usable with decryption session (decryption of 1st parameter of command)
- In “Indicator” column, the value of indicator can be approved, non-approved or non-security relevant.
- In “Inputs” column, commands inputs are not exhaustive, some non-security parameters are voluntarily missed. Full inputs of all commands are defined in **[TPM2.0 Part3]**
- In “Outputs” column, Outputs of all responses are defined in **[TPM2.0 Part3]**
- In “Security Function Implementation” column, security functions are referenced by their identifiers indicated in the tables of the paragraph 2.5.
- In “Roles” column, NA indicates that the service does not require authentication. The list of roles is indicated in paragraph 4.2.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_Init	Reboot or power-on of the TPM.	Non-security relevant	None	None	None	Unauthenticated - nullSeed: Z - nullProof: Z - platformAuth: Z - objSeed: Z - objAuth: Z - objSens: Z - objPub: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- sesSalt: Z - sesHmacKey: Z - sesSymKey: Z - contextKey: Z - drbgSeed: Z - objSymKey: Z - objHmacKey: Z - contextEncKey: Z - dupSeed: Z - dupInSymKey: Z - dupOutSymKey: Z - dupOutHmacKey: Z - creSeed: Z -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						creSymKey: Z - creHmacKey: Z - ephSensEccKey: Z - ephPubEccKey: Z - seqAuth: Z - tdrbgState: Z - fuSymKey: Z - diagSymKey: Z
TPM2_Startup	Set-up the TPM after a power cycle.	Approved	Startup type	None	ENT - ESV DRBG	Unauthenticated - phSeed: G - ehSeed: G - shSeed: G - phProof: G - ehProof: G - shProof:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G - contextKey: G - drbgSeed: G - drbgState: G - nullSeed: G - nullProof: G
TPM2_Shutdown (I)	Prepare the TPM for a power cycle.	Non-security relevant	Shutdown type	None	None	Unauthenticated
TPM2_SelfTest (I)	Self-tests execution	Approved	Full or background self-tests	Self-test result if full self-tests required	AES-ENC AES-DEC SigGen SigVer DRBG ENT-ESV KAS KB KDF MA	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					C SHA	
TPM2_IncrementalSelfTest (I)	Incremental self-tests execution	Approved	List of tests to pass	List of remaining tests	AES-ENC AES-DEC SigGen SigVer DRBG ENT-ESV KAS KB KDF MAC SHA	Unauthenticated
TPM2_GetTestResult (I)	Get self-tests result	Non-security relevant	None	Self-tests status	KB KDF	Unauthenticated - diagSymKey: G,E,Z - diagSymSeed: E
TPM2_StartAuthSession (I/E/D)	Session command	Approved	Decryption key handle; Binding entity handle; Encrypted salt; Nonce	Nonce TPM	KAS KTS-IFC KB KDF	Unauthenticated - sesHmacKey: G,W - sesSymKey: G,W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			caller; Session Type (HMAC or Policy)			- sesSalt: W,E,Z - objSens: E - objAuth: E - nvAuth: E - platformAuth: E - endorsementAuth: E - ownerAuth: E - lockoutAuth: E - seqAuth: E
TPM2_PolicyRestart (I)	Policy session restart	Non-security relevant	Session handle	None	None	Unauthenticated
TPM2_Create (I/E/D)	Object creation	Approved	Parent object handle Object sensitive part Object public	Object private part (encrypted) Object public part Creation data Digest of creation	KeyGen AES - ENC SigGen SigV	User (U) - objSeed: G,R,E - objSymKey: G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			template Creation data List of PCR	data Ticket to be used by TPM2_CertifyCreation( )	er DRBG ENT - ESV KTS KBKDF MAC SHA CKG KAS - KeyGen	- objHmacKey: G,E,Z - objSens: G,R,E - objPub: G,R,E - drbgState: W,E - objAuth: W,R - nullProof: E - phProof: E - ehProof: E - shProof: E
TPM2_Load (I/E/D)	Object loading	Approved	Parent object handle Object private part (encrypted) Object public part	Name of the loaded object	KeyVer AES - DEC KTS KBKDF MAC SHA	User (U) - objSymKey: G,W,E,Z - objHmacKey: G,W,E,Z - objSens: W,E - objPub: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- objSeed: W,E - objAuth: W
TPM2_LoadExternal (I/E/D)	External object loading	Approved	Object public part Hierarchy	Name of the loaded object	Key Val	Unauthenticated - objPub: W - objSens: W - objAuth: W - objSeed: W
TPM2_ReadPublic (I)	Read public part of a loaded object	Approved	Handle of an object	Object public part Object name Object qualified name	None	Unauthenticated - objPub: R
TPM2_ActivateCredential (I/E/D)	Enables the association of a credential with an object in a way that ensures that the TPM has validated the parameters of the credentialed object	Approved	Handle of the object with credentials Handle of a loaded private key Encrypted credential	Decrypted certificate information	AES-DEC KAS KTS-IFC KTS KB KDF MAC SHA	Crypto officer (CO) - creSymKey: G,E,Z - creHmacKey: G,E,Z - objSens: E -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			Encrypted seed			creSeed: W,E,Z
TPM2_MakeCredential (I/E/D)	Allows the TPM to perform the actions required of a Certificate Authority (CA) in creating a TPM2B_ID_OBJECT containing an activation credential	Approved	Handle of a loaded public key Credential information Name of the object with credentials	Encrypted credential Encrypted seed	AES-ENC KAS KTS-IFC KTS KB KDF MAC SHA	Unauthenticated - creSeed: G,R,E,Z - creSymKey: G,E,Z - creHmacKey: G,E,Z - objPub: E
TPM2_Unseal (I/E/D)	Returns the data in a loaded Sealed Data Object	Approved	Handle of a loaded data object	Unsealed data	None	User (U) - objSens: R
TPM2_ObjectChangeAuth (I/E/D)	Changes the authorization secret for a TPM-resident object	Approved	Handle of an object Handle of the parent of the object New authorization value	Object private part	AES-ENC KB KDF MAC SHA	User (U) - objSeed: R,E - objSens: R - drbgState: W,E - objAuth: R - objSymKey: E - objHmacKey: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_CreateLoaded (I/E/D)	Creates an object and loads it in the TPM	Approved	Parent object handle Object sensitive part Object public template	Object private part (encrypted) Object public part Creation object name	Key Gen Key Ver AES - ENC SigGen SigVer DRBG ENT - ESV KAS KB KDF MAC SHA CKG KAS - Key Gen	Crypto officer (CO) - objSeed: G,R,E - objSymKey: G,E - objHmacKey: G,E - objSens: G,R,E - objPub: G,R,E - tdrbgState: G,W,E - drbgState: W,E - objAuth: W,R - nullSeed: E - phSeed: E - ehSeed: E - shSeed: E - nullProof: E - phProof:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - ehProof: E - shProof: E - ekRsa: E - ekEcc: E - shProofForReseed: G,E User (U) - objSeed: G,E - objSymKey: G,E - objHmacKey: G,E - objSens: G,R - objPub: G,R,E - tdrbgState: G,W,E - drbgState: W - objAuth: W - nullSeed:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - phSeed: E - ephSensEccKey: E - shSeed: E - nullProof: E - phProof: E - ehProof: E - shProofForReseed: G,E - ekRsa: E - ekEcc: E
TPM2_Duplicate (I/E/D)	Duplicates a loaded object so that it may be used in a different hierarchy	Approved	Handle of the loaded object to duplicate Handle of the new parent Optional symmetric encryption key	Encryption key for inner wrapper Duplicated object private part (encrypted) Encrypted seed	AES-ENC DRBG KTS-IFC KAS KTS KB KDF MAC SHA	User (U) - dupSeed: G,R,E,Z - objSeed: R - dupOutSymKey: G,E,Z - dupInSymKey:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					CKG	G,R,W,E,Z - dupOutHmacKey: G,E,Z - objSens: R - objAuth: R - drbgState: W,E - objPub: E
TPM2_Rewrap (I/E/D)	Rewraps a duplicated object with a new parent key	Approved	Handle of the old parent Handle of the new parent Duplicated object private part (encrypted) Name of the object being rewrapped Encrypted seed	Duplicated object private part (encrypted) Encrypted seed	AES-ENC AES-DEC KAS KTS-IFC KTS KB KDF MAC SHA CKG	User (U) - dupOutSymKey: G,E,Z - dupOutHmacKey: G,E,Z - objSens: R,W,E - dupSeed: R,W,E,Z - objSeed: R,W - dupInSymKey: W,Z - drbgStat

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						e: W,E - objPub: E - objAuth: R,W
TPM2_Import (I/E/D)	Allows an object to be encrypted using the symmetric encryption values of a Storage Key	Approved	Handle of the new parent Duplicated object private part (encrypted) Object public part Encrypted seed Encryption key for inner wrapper	Object private part (encrypted)	AES-ENC AES-DEC KAS KTS-IFC KTS KBKDF MAC SHA CKG	User (U) - objSens: R,W,E,Z - objSeed: R,W,Z - objPub: W,E,Z - dupOutSymKey: W,E,Z - objAuth: R,W,Z - drbgState: E - dupSeed: E,W,Z - dupInSymKey: E,W,Z - dupOutHmacKey: W,E,Z
TPM2_RSA_Encrypt (I/E/D)	Performs RSA encryption	Approved	RSA public key handle Message	Encrypted output	KTS-IFC	Unauthenticated - objPub: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			to encrypt RSA scheme to use			
TPM2_RSA_Decrypt (I/E/D)	Performs RSA decryption	Approved	RSA private key handle Ciphertext to decrypt RSA scheme to use	Decrypted output	KTS-IFC	User (U) - objSens: Z
TPM2_ECDH_KeyGen (I/E/D)	Shared secret value computation using ECDH	Approved	ECC key public part handle	Shared secret Ephemeral public key	KAS - Key Gen	Unauthenticated - ephSensEccKey: G,E,Z - ephPubEccKey: G,R,Z - drbgState: W,E - objPub: E
TPM2_ECDH_ZGen (I/E/D)	Shared secret value recovery using ECDH	Approved	Handle of a loaded ECC key Ephemeral public key	Recovered shared secret	KAS	User (U) - ephPubEccKey: W,E,Z - objSens: E
TPM2_ECC_Parameters (I)	Returns the parameters of an ECC curve	Non-secu	ID of an ECC curve	Curve parameters	None	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	identified by its TCG-assigned curveID	rity relevant				
TPM2_EncryptDecrypt (I/E)	Symmetric encryption or decryption	Approved	Symmetric key handle Decryption or encryption indicator Input IV Data Mode	Encrypted or decrypted data Output IV (for chaining)	AES - ENC AES - DEC	User (U) - objSens: E
TPM2_EncryptDecrypt 2 (I/E/D)	Symmetric encryption or decryption	Approved	Symmetric key handle Decryption or encryption indicator Input IV Data Mode	Encrypted or decrypted data Output IV (for chaining)	AES - ENC AES - DEC	User (U) - objSens: E
TPM2_Hash (I/E/D)	Performs a hash operation on data	Approved	Data to hash Hash algorithm Hierarchy to use for ticket	Digest Ticket linked to the input hierarchy	MAC SHA	Unauthenticated - nullProof: E - phProof: E - ehProof: E - shProof: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_HMAC (I/E/D)	Performs a HMAC operation on data	Approved	Symmetric signing key handle Data to HMAC Hash algorithm	HMAC	MAC	User (U) - objSens: E
TPM2_GetRandom (I/E)	Outputs random bytes from a DRBG	Approved	Number of random bytes to generate	Output random bytes	DRBG	Unauthenticated - drbgState: W,E
TPM2_StirRandom (I/D)	Reseed the state of a DRBG	Approved	Additional information	None	DRBG ENT - ESV	Unauthenticated - drbgSeed: W,E,Z - drbgState: W,E
TPM2_HMAC_Start (I/D)	Starts an HMAC sequence	Approved	Handle of an HMAC key Authorization value for sequence Hash algorithm	Sequence handle	MAC	User (U) - seqAuth: W - objSens: E
TPM2_HashSequenceStart (I/D)	Starts a hash or an event sequence	Approved	Authorization value for sequence Hash algorithm	Sequence handle	SHA	Unauthenticated - seqAuth: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_SequenceUpdate (I/D)	Adds data to a hash or HMAC sequence	Approved	Sequence handle Data to hash/HMAC	None	MAC SHA	User (U) - objSens: E
TPM2_SequenceComplete (I/E/D)	Adds last part of data to a hash or HMAC sequence and returns the result	Approved	Sequence handle Data to hash/HMAC Hierarchy for ticket	HMAC or digest Ticket linked to the input hierarchy	MAC SHA	User (U) - nullProof: E - phProof: E - ehProof: E - shProof: E - objSens: E - seqAuth: Z
TPM2_EventSequenceComplete (I/D)	Adds last part of data to a hash or HMAC sequence and returns the result in a digest list	Approved	Handle of PCR to extend Sequence handle Data to hash/HMAC	List of digests computed for the PCR	MAC SHA	User (U) - objSens: E - seqAuth: Z
TPM2_Certify (I/E/D)	Proves that an object with a specific Name is loaded in the TPM	Approved	Handle of the object to certify Handle of a signing key Qualifyi	Certification structure Signature over the certification structure	SigGen DRBG KBKDF MAC SHA	User (U) - drbgState: W,E - objSens: E -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			ng data Signature scheme		CKG	shProof: E
TPM2_CertifyCreation (I/E/D)	Proves the association between an object and its creation data	Approved	Handle of the object to certify Handle of a signing key Qualifying data Signature scheme Ticket Creation hash	Certification structure Signature over the certification structure	SigGen DRBG KBKDF MAC SHA CKG	User (U) - drbgState: W,E - objSens: E - nullProof: E - phProof: E - ehProof: E - shProof: E
TPM2_Quote (I/E/D)	Quotes PCR values	Approved	Handle of a signing key Qualifying data Selection of PCRs Signature scheme	Quoted information Signature over the quoted information	SigGen DRBG KBKDF MAC SHA CKG	User (U) - drbgState: W,E - objSens: E - shProof: E
TPM2_GetSessionAuditDigest (I/E/D)	Returns a digital signature of the audit session digest	Approved	Handle of a privacy administrator Handle of a signing	Audit information Signature over the quoted information	SigGen KBKDF DRBG MAC	Crypto officer (CO) - objSens: E - shProof:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			key Handle of an audit session Qualifying data Signature scheme		SHA CKG	E - drbgState: W,E
TPM2_GetCommandAuditDigest (I/E/D)	Returns the current value of the command audit digest, a digest of the commands being audited, and the audit hash algorithm	Approved	Handle of a privacy administrator Handle of a signing key Qualifying data Signature scheme	Audit information Signature over the quoted information	SigGen DRBG KB KDF MAC SHA CKG	Crypto officer (CO) - drbgState: W,E - objSens: E - shProof: E
TPM2_GetTime (I/E/D)	Returns the current values of Time and Clock	Approved	Handle of a privacy administrator Handle of a signing key Qualifying data Signature scheme	Attestation data Signature over the attestation data	SigGen KB KDF DRBG MAC SHA CKG	Crypto officer (CO) - drbgState: W,E - objSens: E - shProof: E
TPM2_CertifyX509 (I/E/D)	X.509 certificate generation	Approved	Handle of the object to certify Handle of a	Additional certificate information Digest Signature	SigGen SHA	User (U) - drbgState: W,E -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			signing key Partial certificate Signature scheme	over the digest		objSens: E
TPM2_VerifySignature (I/D)	Uses loaded keys to validate a signature on a message with the message digest passed to the TPM	Approved	Handle of a public key Digest of a message Signature to be tested	Validation ticket	SigVer MAC	Unauthenticated - objPub: E - nullProof: E - phProof: E - ehProof: E - shProof: E
TPM2_Sign (I/D)	Causes the TPM to sign an externally provided hash with the specified symmetric or asymmetric signing key	Approved	Handle of a signing key Digest to be signed Scheme Proof ticket for digest	Signature over the digest	SigGen DRBG MAC SHA	User (U) - objSens: E - nullProof: E - phProof: E - ehProof: E - shProof: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_SetCommandCodeAuditStatus (I)	Changes the audit status of a command or to set the hash algorithm used for the audit digest	Non-security relevant	Authorization handle Hash algorithm	None	None	Crypto officer (CO)
TPM2_PCR_Extend (I)	Updates the indicated PCR	Approved	PCR handle List of digests used to extend PCRs	None	SHA	Unauthenticated
TPM2_PCR_Event (I/D)	Updates the indicated PCR and reports list of digests	Approved	PCR handle Event data	Digests	SHA	Unauthenticated
TPM2_PCR_Read (I)	Returns the values of all PCR specified in pcrSelectionIn	Non-security relevant	Selection of PCR to read	PCR information	None	Unauthenticated
TPM2_PCR_Allocate (I)	Sets the desired PCR allocation of PCR and algorithms	Non-security relevant	Selection of PCR to allocate	PCR allocation information	None	Crypto officer (CO)
TPM2_PCR_Reset (I)	Sets the PCR in all banks to zero	Non-security relevant	PCR to reset	none	None	Unauthenticated
_TPM_Hash_Start	Indicates to the TPM interface the start of an H-CRTM	Approved	None	None	SHA	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	measurement sequence					
_TPM_Hash_Data	Indicates to the TPM interface data to be included in the H-CRTM measurement sequence	Approved	Data	None	SHA	Unauthenticated
TPM_Hash_End	Indicates to the TPM interface the end of the H-CRTM measurement sequence	Approved	None	None	SHA	Unauthenticated
TPM2_PolicySigned (I/E/D)	Includes a signed authorization in a policy	Approved	Signature key handle Policy session handle Nonce TPM Digest Signature Expiration of authorization Policy reference value	Policy timeout Policy ticket	SigVer MAC SHA	Unauthenticated - objPub: E - nullProof: E - phProof: E - ehProof: E - shProof: E
TPM2_PolicySecret (I/E/D)	Includes a secret-based authorization to a policy	Approved	Authorization object handle Policy session handle Nonce	Policy timeout Policy ticket	MAC SHA	User (U) - nullProof: E - phProof: E -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			TPM Digest Expiration of authorization Policy reference value			ehProof: E - shProof: E
TPM2_PolicyTicket (I/D)	Includes a ticket in a policy	Approved	Policy session handle Nonce TPM Digest Expiration of authorization Policy reference value Authorization object name Ticket	None	MAC SHA	Unauthenticated - nullProof: E - phProof: E - ehProof: E - shProof: E
TPM2_PolicyOR (I)	Allows options in authorizations without requiring that the TPM evaluate all the options	Approved	Policy session handle List of digests	None	SHA	Unauthenticated
TPM2_PolicyPCR (I/D)	Causes conditional gating of a policy based on PCR	Approved	Policy session handle Expected digest value	None	SHA	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			PCR selection			
TPM2_PolicyLocality (I)	Indicates that the policy will be limited to a specific locality	Approved	Policy session handle Locality	None	SHA	Unauthenticated
TPM2_PolicyNV (I/D)	Causes conditional gating of a policy based on the contents of an NV Index	Approved	Authorization handle NV index handle Policy session handle Operand, offset, operation	None	SHA	User (U)
TPM2_PolicyCounterTimer (I/D)	Causes conditional gating of a policy based on the contents of the TPMS_TIME_INFO structure	Approved	Policy session handle Operand, offset, operation	None	SHA	Unauthenticated
TPM2_PolicyCommandCode (I)	Limits policy to a specific command code	Approved	Policy session handle Command code	None	SHA	Unauthenticated
TPM2_PolicyPhysicalPresence (I)	Physical presence will need to be asserted at the time the authorization is performed	Approved	Policy session handle	None	SHA	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_PolicyCpHash (I/D)	Allows a policy to be bound to a specific command and command parameters	Approved	Policy session handle Digest to add to policy	None	SHA	Unauthenticated
TPM2_PolicyNameHash (I/D)	Allows a policy to be bound to a specific set of TPM entities without being bound to the parameters of the command	Approved	Policy session handle Digest to add to policy	None	SHA	Unauthenticated
TPM2_PolicyDuplicationSelect (I/D)	Allows qualification of duplication to allow duplication to a selected new parent	Approved	Policy session handle Object name to be duplicated New Parent name Object name inclusion indicator	None	SHA	Unauthenticated
TPM2_PolicyAuthorize (I/D)	Check a ticket issued from the signature verification of a new policy so that it may be used in an existing policy	Approved	Policy session handle Digest of the policy being approved Policy qualifier Key name Ticket	None	MAC SHA	Unauthenticated - nullProof: E - phProof: E - ehProof: E -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						shProof: E
TPM2_PolicyAuthValue (I)	Allows a policy to be bound to the authorization value of the authorized entity	Approved	Policy session handle	None	SHA	Unauthenticated
TPM2_PolicyPassword (I)	Allows a policy to be bound to the authorization value of the authorized object	Approved	Policy session handle	None	SHA	Unauthenticated
TPM2_PolicyGetDigest (I/E)	Returns the current policyDigest of a policy session	Non-security relevant	Policy session handle	Policy digest	None	Unauthenticated
TPM2_PolicyNvWritten (I)	Allows a policy to be bound to the TPMA_NV_WRITTEN attributes	Approved	Policy session handle NV index written indicator	None	SHA	Unauthenticated
TPM2_PolicyTemplate (I/D)	Allows a policy to be bound to a specific creation template	Approved	Policy session handle Digest to add to policy	None	SHA	Unauthenticated
TPM2_PolicyAuthorizeNV (I)	Provides a capability that is the equivalent of a revocable policy	Approved	Source handle for authorization NV index to	None	SHA	User (U)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			read Policy session handle			
TPM2_CreatePrimary (I/E/D)	Creates a Primary Object under one of the Primary Seeds or a Temporary Object under TPM_RH_NULL	Approved	Primary handle Key sensitive data Key public template Creation data Creation PCR	Object handle Object Public part Creation data Digest of creation data Creation ticket Name of the object	Key Gen Key Ver SigGen SigVer DRBG KBKDF MAC SHA CKG KAS - Key Gen	Crypto officer (CO) - objSeed: G,E,Z - objSymKey: G,E,Z - objHmacKey: G,E,Z - objSens: G,E,Z - objPub: G,R,E,Z - tdrbgState: G,W,E,Z - drbgState: W,E - objAuth: W - nullSeed: E - phSeed: E - ehSeed: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- shSeed: E - nullProof: E - phProof: E - ehProof: E - shProof: E - ekRsa: E - ekEcc: E - shProofForReseed: G,E
TPM2_HierarchyControl (I)	Enables and disables use of a hierarchy and its associated NV storage	Non-security relevant	Primary handle Hierarchy to enable or disable Enable or disable indicator	None	None	Crypto officer (CO)
TPM2_SetPrimaryPolicy (I/D)	Sets the authorization policy for a hierarchy	Non-security relevant	Primary handle Policy digest Hash algorithm	None	None	Crypto officer (CO)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_ChangePPS (I)	Replaces the current platform primary seed (PPS) with a value from the RNG and sets platformPolicy to the default initialization value	Approved	Authorization handle	None	DRBG	Crypto officer (CO) - drbgState: W,E - phProof: Z - phSeed: Z - objSeed: Z - objSens: Z - objPub: Z
TPM2_ChangeEPS (I)	Replaces the current endorsement primary seed EPS) with a value from the RNG and sets endorsementPolicy to the default initialization value	Approved	Authorization handle	None	DRBG	Crypto officer (CO) - drbgState: W,E - ehSeed: Z - ehProof: Z - objSeed: Z - objSens: Z - objPub: Z - ekRsa: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- ekEcc: Z
TPM2_Clear (I)	Removes all TPM context associated with a specific Owner	Approved	Authorization handle	None	DRBG	Crypto officer (CO) - drbgState: W,E - shSeed: Z - ehProof: Z - shProof: Z - shProofForReseed: Z - objSeed: Z - objSens: Z - objPub: Z - objAuth: Z
TPM2_ClearControl (I)	Disables and enables the execution of TPM2_Clear()	Non-security relevant	Authorization handle Set or clear disableOwnerFlag	None	None	Crypto officer (CO)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_HierarchyChangeAuth (I/D)	Changes the authValue of hierarchies	Non-security relevant	Authorization handle New authorization value	None	None	Crypto officer (CO) - lockoutAuth: W - endorsementAuth: W - ownerAuth: W - platformAuth: W
TPM2_DictionaryAttackLockReset (I)	Cancels the effect of a TPM lockout due to several successive authorization failures	Non-security relevant	Authorization handle	None	None	Crypto officer (CO)
TPM2_DictionaryAttackParameters (I)	Changes the lockout parameters	Non-security relevant	Authorization handle newMaxTries, newRecoveryTime and lockoutRecovery values	None	None	Crypto officer (CO)
TPM2_VendorCmdFieldUpgradeStart (I)	Initiates a field upgrade session	Approved	Approved	None	SigVer KBKDF SHA CKG	Crypto officer (CO) - fuSigECCKey: E -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						fuSigLMSKey: E - fuSymKey: G - fuSymSeed: E
TPM2_VendorCmdFieldUpgradeData (I)	Conveys firmware in a field upgrade session	Approved	Field upgrade data blob	Completion indicator	AES - DEC SHA	Unauthenticated - fuSymKey: E,Z
TPM2_ContextSave	Saves a session context, object context, or sequence object context outside the TPM	Approved	Saved handle	Context	AES - ENC KTS KB KDF MAC CKG	Unauthenticated - contextEncKey: G,E,Z - objSeed: R - objSens: R - objPub: R - objAuth: R - nullProof: E - phProof: E - ehProof: E - shProof:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - sesHmacKey: R - seqAuth: R - contextKey: E
TPM2_ContextLoad	Reloads a context that has been saved by TPM2_ContextSave()	Approved	Context	Loaded handle	AES - DECKTS KBKDF MAC CKG	Unauthenticated - contextEncKey: G,E,Z - objSeed: W - objSens: W - objPub: W - objAuth: W - nullProof: E - phProof: E - ehProof: E - shProof: E - sesHmac

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key: W - seqAuth: W - contextKey: E
TPM2_FlushContext	Causes all context associated with a loaded object, sequence object, or session to be removed from TPM memory	Approved	Flush handle	None	None	Unauthenticated - objSeed: Z - objSens: Z - objPub: Z - objAuth: Z
TPM2_EvictControl (I)	Allows certain Transient Objects to be made persistent or a persistent object to be evicted	Approved	Authorization handle Loaded object handle Persistent handle	None	None	Crypto officer (CO) - objSeed: W,Z - objSens: W,Z - objPub: W,Z - objAuth: W,Z - sesHmacKey: W - sesSymKey: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_ReadClock (I)	Reads the current TPMS_TIME_INFO structure	Non-security relevant	None	Current time	None	Unauthenticated
TPM2_ClockSet (I)	Advances the value of the TPM's clock	Non-security relevant	New time	None	None	Crypto officer (CO)
TPM2_ClockRateAdjust (I)	Adjusts the rate of advance of Clock and Time	Non-security relevant	Authorization handle Clock update rate adjustment	None	None	Crypto officer (CO)
TPM2_GetCapability (I)	Returns various information regarding the TPM and its current state	Non-security relevant	Capability, property, property count	More data availability indicator Capability data	None	Unauthenticated
TPM2_SetCapability (I/D)	Set specific data in the TPM, such as TPM configurations, which may change the TPM's function and behavior	Non-security relevant	Capability data	None	None	Crypto officer (CO)
TPM2_TestParms (I)	Checks if specific combinations of algorithm parameters are supported	Non-security relevant	Algorithm parameters	None	None	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_NV_DefineSpace (I/D)	Defines the attributes of an NV Index and causes the TPM to reserve space to hold the data associated with the NV Index	Approved	Authorization handle NV authorization value NV public parameters	None	None	Crypto officer (CO) - nvAuth: W
TPM2_NV_UndefineSpace (I)	Removes an Index from the TPM	Approved	Authorization handle NV index to delete	None	None	Crypto officer (CO) - nvAuth: Z
TPM2_NV_UndefineSpaceSpecial (I)	Removal of a platform-created NV Index that has TPMA_NV_POLICY_DELETE SET	Approved	Platform authorization handle NV index to delete	None	None	Crypto officer (CO) - nvAuth: Z
TPM2_NV_ReadPublic (I/E)	Reads the public area and Name of an NV Index	Approved	NV index	NV index public area Name of the NV index	SHA	Unauthenticated
TPM2_NV_Write (I/D)	Writes a value to an area in NV memory that was previously defined by TPM2_NV_DefineSpace()	Non-security relevant	Authorization handle NV index to write Data to write Offset in the NV index area	None	None	User (U)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_NV_Increment (I)	Increments the value in an NV Index that has the TPM_NT_COUNTER attribute	Non-security relevant	Authorization handle NV index to increment	None	None	User (U)
TPM2_NV_Extend (I/D)	Extends a value to an area in NV memory that was previously defined by TPM2_NV_DefineSpace()	Approved	Authorization handle NV index to extend Data to extend	None	SHA	User (U)
TPM2_NV_SetBits (I)	Sets bits in an NV Index that was created as a bit field	Non-security relevant	Authorization handle NV index to extend Data to OR with NV content	None	None	User (U)
TPM2_NV_WriteLock (I)	Inhibits further writes of the NV Index if the TPMA_NV_WRITEDEFINE or TPMA_NV_WRITE_STCLEAR attributes of an NV location are SET	Non-security relevant	Authorization handle NV index	None	None	User (U)
TPM2_NV_GlobalWriteLock (I)	Sets TPMA_NV_WRITELOCKED for all indexes that have their	Non-security	Authorization handle	None	None	Crypto officer (CO)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	TPMA_NV_GLOBALLOCK attribute SET	relevant				
TPM2_NV_Read (I/E)	Reads a value from an area in NV memory previously defined by TPM2_NV_DefineSpace()	Non-security relevant	Authorization handle NV index to be read Size and offset in NV area	Data read	None	User (U)
TPM2_NV_ReadLock (I)	Prevents further reads of the NV Index until the next TPM2_Startup (TPM_SU_CLEAR) if TPMA_NV_READ_STCLEAR is SET	Non-security relevant	Authorization handle NV index to be locked	None	None	User (U)
TPM2_NV_ChangeAuth (I/D)	Allows the authValue of an NV Index to be changed	Approved	NV index New authorization value	None	None	User (U) - nvAuth: W
TPM2_NV_Certify (I/E/D)	Certifies the contents of an NV Index or portion of an NV Index	Approved	Handle of signing key Authorization handle NV index Qualifying data Scheme Size and	Structure that was signed Signature	SigGen KBKDF MAC SHA	User (U) - objSens: E - shProof: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			offset in NV area			
TPM2_VendorCmdSetMode (I)	Sets the low power mode	Non-security relevant	Authorization handle Low power configuration structure	None	None	Crypto officer (CO)
TPM2_VendorCmdSetCommandSet (I)	Activates and locks commands	Non-security relevant	Authorization handle Command code Activation and lock indicators	None	None	Crypto officer (CO)
TPM2_VendorCmdSetCommandSetLock (I)	Prevents locking commands	Non-security relevant	Authorization handle	None	None	Crypto officer (CO)
TPM2_VendorCmdGetRandom2 (I/E)	Get random value from DRBG	Approved	Number of bytes to generate	Random value	None	Unauthenticated - drbgState: W,E
TPM2_VendorCmdGPIOConfig (I)	Configures GPIO	Non-security relevant	Authorization handle GPIO configuration	None	None	Unauthenticated
TPM2_VendorCmdGetRandom800_90B (I/E)	Get random value from ESV Cert. #E41	Approved	Number of bytes to generate	Random value	ENT - ESV	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TPM2_VendorCmdChangeObjectDeletionAuth (I)	Modifies deletion authorization for an object	Non-security relevant	Authorization handle Platform authorization use indicator	None	None	Crypto officer (CO)
TPM2_VendorCmdRestoreEK (I)	Restore EK RSA or EK ECC in case of deletion by TPM2_ChangeEPS	Approved	Authorization handle	None	None	Crypto officer (CO) - ekRsa: W - ekEcc: W
TPM2_VendorCmdZeroizeEK (I)	Zeroize EK RSA and EK ECC	Approved	Authorization handle	None	None	Crypto officer (CO) - ekRsa: Z - ekEcc: Z
TPM2_VendorCmdSetBackgroundSlotsConfig	Configure the RSA background key slots	Non-security relevant	Authorization handle Slots configuration	None	None	Crypto officer (CO)
TPM2_PP_Commands	Determines which commands require assertion of Physical Presence	Non-security relevant	Authorization handle List of commands to add and list of command to remove	None	None	Crypto officer (CO)
Integrity mechanism provided by sessions	This service is not callable from TPM	Approved	Command or response	Integrity value	DRBG KB	Unauthenticated -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	interface but is only used internally by any command and response with an authorization area. It consists in computing the integrity of the received command or transmitted response.				KDF MAC SHA CKG	sesHmacKey: E,Z
Encryption mechanism provided by sessions	This service is not callable from TPM interface but is only used internally by any command and response with an encryption or decryption session. It consists in decrypting the first parameter of a received command or encrypting the first parameter of a transmitted response.	Approved	Command or response	Encrypted parameter	AES-ENC AES-DEC DRBG KB KDF SHA CKG	Unauthenticated - sesSymKey: G,E,Z

Table 23: Approved Services

The integrity mechanism provided by sessions is not directly callable from the security module external interfaces. Function is used (or might be used) by the services listed in this table. When a service is usable with a session, (I) is added next to the service name. When a service can additionally use the encryption mechanism of a session, (I/E) is added next to the service name.

The encryption mechanism provided by sessions is not directly callable from the security module external interfaces. Function is used (or might be used) by the services listed in this table. When a service is usable with a session, (I) is added next to the service name. When a service can additionally use the encryption mechanism of a session, (I/E) is added next to the service name.

4.4 Non-Approved Services

All approved services implemented by the Module are listed in the table below:

Name	Description	Algorithms	Role
TPM2_Create; TPM2_CreateLoaded; TPM2_Load; TPM2_LoadExternal	Creation or loading of an ECC key with a non-approved elliptic curve; Creation or loading of an ECC key for a non-approved key agreement usage; Creation or loading of an ECC signing key with an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme = TPM_ALG_NULL);Creation or loading of an RSA decryption key with an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme = TPM_ALG_NULL); Creation or loading of a 1024-bit RSA key	ECC BN P-256 (non-compliant) RSA (non-compliant)	User
TPM2_CreateLoaded	Derivation of an ECC key from a derivation parent key	ECC derived keys (non-compliant) KBKDF (non-compliant)	User
TPM2_Load; TPM2_LoadExternal	Loading of an ECC or RSA key (sensitive and public parts) in the NULL hierarchy	ECC BN P-256 (non-compliant) RSA (non-compliant)	User
TPM2_Duplicate; TPM2_Rewrap; TPM2_Import	Key transport with a 1024-bit RSA key Key agreement scheme with a non-approved ECC curve Key agreement scheme with an ECC key used in a non-approved key agreement usage	ECC BN P-256 (non-compliant) KAS (non-compliant) RSA (non-compliant)	User

Name	Description	Algorithms	Role
TPM2_RSA_Encrypt; TPM2_RSA_Decrypt	Key transport with a non-approved scheme: * RSAES-PKCS1-v1_5 * RSA with no padding mode (null scheme) Key transport with an RSA decryption key: * Generated with an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme = TPM_ALG_NULL) * Loaded in the NULL hierarchy	KTS-IFC (non-compliant) RSA with no padding mode (null scheme) (non-compliant) RSAES-PKCS1-v1_5 (non-compliant)	User
TPM2_ECDH_KeyGen	Use of a non-approved elliptic curve: * ECC key with curve BN P-256 Use of an ECC key for a non-approved key agreement usage: * ECC key with curve Curve448	ECC BN P-256 (non-compliant)	N/A
TPM2_ECDH_ZGen	Use of an ECC key: * Generated on curve BN P-256 * For a non-approved key agreement usage * Derived from a derivation parent key * Loaded in the NULL hierarchy	ECC BN P-256 (non-compliant) KBKDF (non-compliant)	User
TPM2_ZGen_2Phase	This command is only usable jointly with TPM2_EC_Ephemeral service that is non approved as using key derivation to generate ECC keys	ECC derived keys (non-compliant) KBKDF (non-compliant)	User
TPM2_HMAC	HMAC generation with a key length < 112 bits	HMAC (non-compliant)	User

TPM2_HMAC_Start;
TPM2_SequenceUpdate;
TPM2_SequenceComplete
	HMAC generation with a key length < 112 bits	HMAC (non-compliant)	User

TPM2_Certify; TPM2_CertifyCreation; TPM2_Quote;	Digital signature with a non-approved signature scheme: * ECC signature with ECDAA signature scheme * ECC	ECC BN P-256 (non-	User/CO

Name	Description	Algorithms	Role
TPM2_GetSessionAuditDigest; TPM2_GetCommandAuditDigest; TPM2_GetTime; TPM2_CertifyX509	signature with ECSchnorr signature scheme * RSA signature with key length of 1024 bits * ECC or RSA signature key using SHA-1 as digest method * ECC signature with curve BN P-256 Digital signature with an ECC signing key generated with an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme = TPM_ALG_NULL); Digital signature with an ECC signing derived from a derivation parent key; Digital signature with an ECC or RSA key loaded in the NULL hierarchy	compliant) ECDAA (non-compliant) ECDSA (non-compliant) ECSchnorr (non-compliant) RSA (non-compliant) SHA-1 (non-compliant)	
TPM2_Commit	Generation of an ECC key through key derivation method	KBKDF (non-compliant)	User
TPM2_EC_Ephemeral	Generation of an ECC key through key derivation method	KBKDF (non-compliant)	User
TPM2_VerifySignature	Digital signature verification with a non-approved signature scheme or a non-approved curve: * ECDAA signature scheme * ECSchnorr signature scheme * ECC signature with curve BN P-256	ECC BN P-256 (non-compliant) ECDAA (non-compliant) ECSchnorr (non-compliant)	NA
TPM2_Sign	Digital signature generation with a non-approved signature scheme: * ECC signature with ECDAA signature scheme * ECC signature with ECSchnorr signature scheme * RSA signature with key length of 1024 bits * ECC or RSA signature key using SHA-1 as digest method * ECC signature with curve BN P-256; Digital signature with an ECC signing key generated with an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme =	ECC BN P-256 (non-compliant) ECDAA (non-compliant) ECDSA (non-compliant) ECSchnorr	User

Name	Description	Algorithms	Role
	TPM_ALG_NULL); Digital signature with an ECC signing derived from a derivation parent key; Digital signature with an ECC or RSA key loaded in the NULL hierarchy	(non-compliant) RSA (non-compliant) SHA-1 (non-compliant)	
TPM2_PolicySigned	Digital signature verification with a non-approved signature scheme or a non-approved curve: * ECDAA signature scheme * ECSchnorr signature scheme * ECC signature with curve BN P-256	ECC BN P-256 (non-compliant) ECDAA (non-compliant) ECSchnorr (non-compliant)	N/A
TPM2_CreatePrimary	Creation and loading of an ECC key with a non-approved elliptic curve: * ECC key with curve BN P-256 Use of an ECC key for a non-approved key agreement usage: * ECC key with curve Curve448 Creation and loading of an ECC signing key with an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme = TPM_ALG_NULL) Creation and loading of an RSA decryption key with an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme = TPM_ALG_NULL)	ECC BN P-256 (non-compliant)	CO
TPM2_NV_Certify	Digital signature with a non-approved signature scheme: * ECC signature with ECDAA signature scheme * ECC signature with ECSchnorr signature scheme * RSA signature with key length of 1024 bits * ECC or RSA signature key using SHA-1 as digest method * ECC signature with curve BN P-256 Digital signature with an ECC signing key generated with an undetermined scheme (field inPublic.buffer.parameters.scheme.scheme =	ECC BN P-256 (non-compliant) ECDAA (non-compliant) ECDSA (non-compliant) ECSchnorr	User

Name	Description	Algorithms	Role
	TPM_ALG_NULL) Digital signature with an ECC signing derived from a derivation parent key Digital signature with an ECC or RSA key loaded in the NULL hierarchy	(non-compliant) RSA (non-compliant) SHA-1 (non-compliant)	

Table 24: Non-Approved Services

4.5 External Software/Firmware Loaded

Loading of firmware on the Module can be achieved by using two services:

- TPM2_VendorCmdFieldUpgradeStart that performs the software/firmware load test detailed in the self-test section of this document to determine if the authorizations to start a loading session are granted
- TPM2_VendorCmdFieldUpgradeData that transports the protected (confidentiality and integrity) parts of the firmware. Several commands are necessary to transport the full firmware.

Data outputs are inhibited until the loading session has completed successfully. Execution of the successfully loaded firmware is only effective after the next reset of the security module.

New firmware versions must be validated through the FIPS 140-3 validation process. Any other firmware loaded into this module is out of the scope of this validation and requires a separate FIPS 140-3 validation.

The core memory loader (CML) represented in Figure 4 is non-modifiable, only the TPM instances are modifiable by using an authenticated firmware upgrade mechanism. The Module contains two instances of the firmware but only one instance is executed after a boot sequence.

5 Software/Firmware Security

5.1 Integrity Techniques

The Module is composed of the following firmware component(s):

- Core Memory Loader executable (aka CML)
- TPM application instance 1 executable
- TPM application instance 2 executable

The firmware integrity is verified by computing a CRC-16 [ISO 13239] over the active firmware and comparing it to a reference value. Firmware integrity is verified during the boot sequence before the execution of the code blocks (CML and TPM). If a failure is detected during the boot sequence, the TPM enters an infinite reset loop that can be exit only by a power-off/power-on sequence. If a failure is detected during the self-tests execution, the security Module enters failure mode.

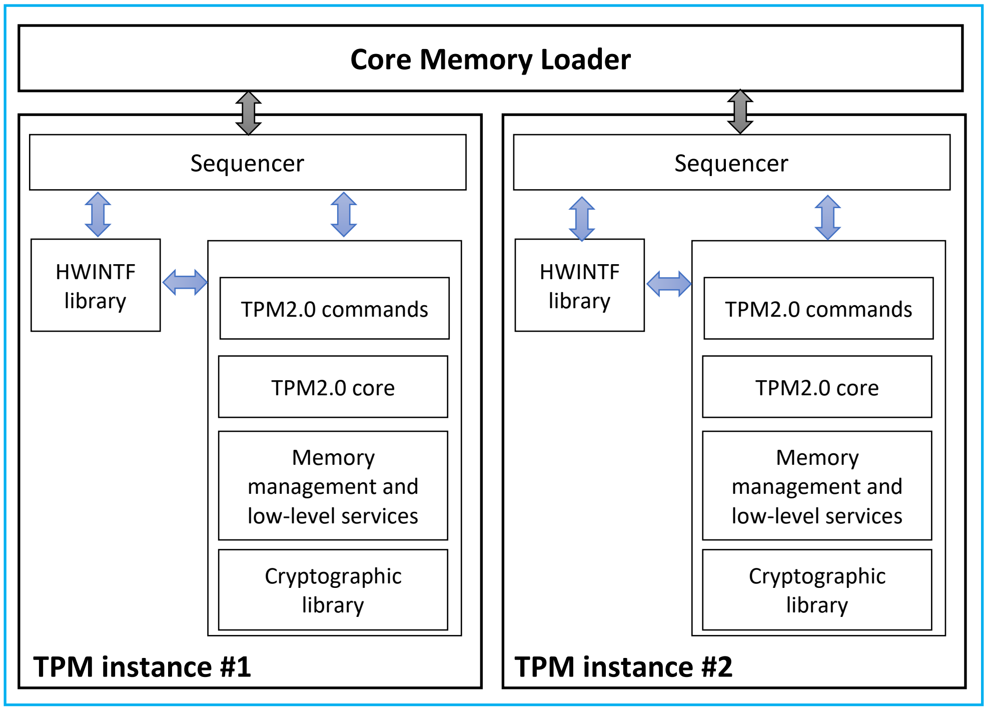

Figure 4 – Firmware block diagram

5.2 Initiate on Demand

The operator can initiate the integrity test on demand by using the TPM2_SelfTest command with the full parameter set to YES or by using the TPM2_IncrementalSelfTest command.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

The operational environment of the Module is “limited” because it allows loading authenticated firmware that meets all applicable requirements of [140-3] standard.

Data outputs are inhibited until the loading session has completed successfully. Execution of the successfully loaded FW is only effective after the next reset of the security module.

New firmware versions must be validated through the FIPS 140-3 validation process. Any other firmware loaded into this module is out of the scope of this validation and requires a separate FIPS 140-3 validation.

The core memory loader (CML) represented in Figure 4 is non-modifiable, only the TPM instances are modifiable by using an authenticated firmware upgrade mechanism.

The security module contains two instances of the FW but only one FW instance is executed after a boot sequence.

7 Physical Security

The security module is production grade and meets the Physical Security protection requirements for single-chip module at FIPS 140-3 Level 3.

7.1 Mechanisms and Actions Required

Zeroisation

Zeroisation of CSPs can be triggered by specific services as detailed in Section 9.3. It occurs in a sufficiently small time-period to prevent the recovery of the sensitive data between start of zeroisation and the zeroisation completeness.

Physical security mechanisms

The security module is encapsulated in a hard opaque package to prevent direct observation of internal security components. It implements additional security mechanisms:

- An active metal shield, located inside the package and covering the internal circuitry and the memory components. Cutting, removing, or modifying the shield layer will cause the security module to reset and enter a shutdown mode.
- An internal circuitry detecting environmental conditions outside the nominal operating range. Power supply voltage and temperature are continuously monitored. If conditions exist outside the range determined by the tamper detection circuitry, the security module resets and enters a failure mode. The module remains in failure mode as long as the environmental condition causing the tamper event persists.

Physical security inspection

Mechanism	Inspection Frequency	Inspection Guidance
Hard opaque package	Dependent on the security module integration environment varies from once per month to once per year	Visual inspection of the package to confirm that it has not been damaged by an external action

Table 25: Mechanisms and Actions Required

7.2 User Placed Tamper Seals

N/A

7.3 Filler Panels

N/A

7.4 Fault Induction Mitigation

N/A

7.5 EFP/EFT Information

EFT has been performed for all security module configurations. Low and high temperatures have been measured at a nominal voltage of 3.3V. Low and high voltage have been measured at ambient temperature (25°C).

The nominal operating ranges are:

- Between 1.62V and 3.8V for voltage
- Between -40°C and +125°C for temperature

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature	-60°C (ST33KTPM2XSPI) / -70°C (ST33KTPM2X)	EFT	Shutdown
HighTemperature	145°C (ST33KTPM2XSPI) / 145°C (ST33KTPM2X)	EFT	Shutdown
LowVoltage	1.5V (ST33KTPM2XSPI) / 1.4V (ST33KTPM2X)	EFT	Shutdown
HighVoltage	4.3V (ST33KTPM2XSPI) / 4.3V (ST33KTPM2X)	EFT	Shutdown

Table 26: EFP/EFT Information

7.6 Hardness Testing Temperature Ranges

Hardness testing was conducted at the temperature indicated in the table below.

Temperature Type	Temperature
LowTemperature	-40°C
HighTemperature	105°C

Table 27: Hardness Testing Temperatures

8 Non-Invasive Security

8.1 Mitigation Techniques

The Module does not claim support of non-invasive attack mitigation techniques referenced in [140F].

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
Dynamic RAM	Volatile memory used to store SSPs between two consecutive resets or power-on/power-off sequence of the security module. SSPs don't persist after command execution. This area is marked as RAM on the HW block diagram.	Dynamic
Static RAM	Volatile memory used to store SSPs between two consecutive resets or power-on/power-off sequence of the security module. SSPs persist after command execution. This area is marked as RAM on the HW block diagram.	Static
NVRAM	Non-volatile memory (flash-based) used to store SSPs and make them persistent to a reset or a power-off/power-on sequence of the security module. This area is marked as flash memory on the HW block diagram.	Static

Table 28: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Input plaintext to NVRAM	Outside of cryptographic boundary	NVRAM	Plaintext	Manual	Electronic	
Input protected to NVRAM	Outside of cryptographic boundary	NVRAM	Encrypted	Manual	Electronic	KTS
Input plaintext to RAM	Outside of cryptographic boundary	Static RAM	Plaintext	Manual	Electronic	
Input protected to RAM	Outside of cryptographic boundary	Static RAM	Encrypted	Manual	Electronic	KTS
Output plaintext	NVRAM	Outside of cryptographic boundary	Plaintext	Manual	Electronic	

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
from NVRAM						
Output protected from NVRAM	NVRAM	Outside of cryptographic boundary	Encrypted	Manual	Electronic	KTS
Output plaintext from RAM	Static RAM	Outside of cryptographic boundary	Plaintext	Manual	Electronic	
Output protected from RAM	Static RAM	Outside of cryptographic boundary	Encrypted	Manual	Electronic	KTS
Input asym. encrypted to RAM	Outside of cryptographic boundary	Static RAM	Encrypted	Manual	Electronic	KTS-IFC
Output asym. encrypted to RAM	Static RAM	Outside of cryptographic boundary	Encrypted	Manual	Electronic	KTS-IFC

Table 29: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
TPM2_Init	Zeroization of all volatile SSPs. Explicit zeroization indicator provided by service completion status.	N/A	Activation of reset signal
TPM2_Clear	Zeroization of all contexts associated with an Owner. Explicit zeroization indicator provided by service completion status.	SSPs linked to an Owner must not persist if the Owner changes	Send TPM2_Clear command
TPM2_Startup	Zeroization of platformAuth. Explicit zeroization indicator provided by service completion status.	Zeroize platformAuth before its first use after a reset	Send TPM2_Startup command
TPM2_ChangePPS	Zeroize the platform primary seed and flush all transient and persistent objects in the Platform hierarchy. Explicit zeroization indicator provided by service completion status.	Platform hierarchy renewal	Send TPM2_ChangePPS command
TPM2_ChangeEPS	Zeroize the endorsement primary seed and flush all transient and persistent objects in the Endorsement hierarchy. Explicit zeroization indicator provided by service completion status.	Endorsement hierarchy renewal	Send TPM2_ChangeEPS command
TPM2_EvictControl	Zeroize an object from NVRAM. Explicit zeroization indicator provided by service completion status.	Method required to zeroize a dedicated object in NVRAM	Send TPM2_EvictControl command
TPM2_FlushContext	Zeroize an object from RAM. Explicit zeroization indicator provided by service completion status.	Method required to zeroize a dedicated object in RAM	Send TPM2_FlushContext command
Automatic	Zeroize SSPs at the end of a command processing. Implicit zeroization indication.	Method for limited life cycle SSPs	No, zeroization is automatic.
TPM2_NV_UndefineSpace TPM2_NV_UndefineSpaceSpecial	Zeroize a NV index. Explicit zeroization indicator provided by service completion status.	Method required to flush NV indices from NVRAM	Send TPM2_NV_UndefineSpace command. Send TPM2_NV_UndefineSpaceSpecial command
TPM2_VendorCmdZeroizeEK	Zeroize the endorsement key provisioned. Explicit zeroization indicator provided by service completion status.	Mandatory zeroization method for EK SSPs	Send TPM2_VendorCmdZeroizeEK command
TPM2_SequenceComplete TPM2_EventSequenceComplete	Zeroize a hash or HMAC sequence. Explicit zeroization indicator provided by service completion status.	Method required to flush sequences from RAM	Send TPM2_SequenceComplete command. Send TPM2_EventSequenceComplete command

Table 30: SSP Zeroization Methods

9.4 SSPs

All usage of these SSPs by the Module are described in the services detailed in section 4. Next table lists the SSPs used as keys:

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
nullProof	Proof (secret value) of the null hierarchy	512 - 256	Symmetric key - CSP	DRBG		KBKDF MAC
phProof	Proof (secret value) of the platform hierarchy	512 - 256	Symmetric key - CSP	DRBG		MAC
ehProof	Proof (secret value) of the endorsement hierarchy	512 - 256	Symmetric key - CSP	DRBG		MAC
shProof	Proof (secret value) of the storage hierarchy	512 - 256	Symmetric key - CSP	DRBG		KBKDF MAC
shProofForReseed	Random value	512 - 256	Entropy source - CSP	ENT-ESV		DRBG
platformAuth	Authentication value for the platform hierarchy	512 - 128 to 256 (depending on the underlying hash algorithm used)	Authentication value / Symmetric key - CSP			KBKDF MAC
endorsementAuth	Authentication value for the endorsement hierarchy	512 - 128 to 256 (depending on the underlying hash algorithm used)	Authentication value / Symmetric key - CSP			KBKDF MAC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ownerAuth	Authentication value for the storage hierarchy	512 - 128 to 256 (depending on the underlying hash algorithm used)	Authentication value / Symmetric key - CSP			KBKDF MAC
lockoutAuth	Authentication value for the lockout hierarchy	512 - 128 to 256 (depending on the underlying hash algorithm used)	Authentication value / Symmetric key - CSP			KBKDF MAC
objSeed	Seed value for object generation	512 - 128 to 256	Data, Symmetric key - CSP	DRBG KBKDF		KBKDF SHA
objAuth	Object's authorization value	112 to 512 - 112 to 256	Authentication value / Symmetric key - CSP			KBKDF MAC
objSymKey	Encryption key of object private part	256 - 256	Symmetric key - CSP	KBKDF		AES-ENC AES-DEC
objHmacKey	Integrity key of object private part	160, 256, 384, 512 - 128 to 256	Symmetric key - CSP	KBKDF		MAC
objSens	Object private part	2048, 3072, 4096 (RSA) 128, 192, 256 (AES) 256, 384, 521 (ECC) 112 to 1024 (HMAC) - 112 to 256	Symmetric or asymmetric private key - CSP	KeyGen KBKDF CKG KAS-KeyGen		AES-ENC AES-DEC SigGen KAS KBKDF MAC
objPub	Object public part	2048, 3072, 4096 (RSA) 2*256, 2*384, 2*521 (ECC) - 112 to 256	Asymmetric public key - PSP	KeyGen KAS-KeyGen		SigVer KAS KTS-IFC
nvAuth	Authorization of NV index	112 to 512 - 112 to 256	Authentication value / Symmetric key - CSP		KTS	KBKDF MAC
sesSalt	Salt for keys diversification	160, 256, 384, 512 - 128 to 256	Symmetric key - CSP	N/A	KAS	KBKDF
sesHmacKey	HMAC session key	160, 256, 384, 512 - 128 to 256	Symmetric key - CSP	KBKDF		KBKDF MAC
sesSymKey	Encrypted session key	128, 192, 256 - 128 to 256	Symmetric key - CSP	KBKDF		AES-ENC AES-DEC
contextKey	Derivation key for context protection	128 - 128	Symmetric key - CSP	DRBG		KBKDF
contextEncKey	Wrapping key for context protection	256 - 256	Symmetric key - CSP	KBKDF		AES-ENC AES-DEC
dupInSymKey	Wrapping key for duplicated object	128, 192, 256 - 128 to 256	Symmetric key - CSP	DRBG		AES-ENC AES-DEC
dupSeed	Seed for protection keys derivation	160 to 512 - 128 to 256	Symmetric key - CSP	DRBG KAS	KAS	KBKDF
dupOutSymKey	Encryption key for duplicated objects	128, 192, 256 - 128 to 256	Symmetric key - CSP	KBKDF		AES-ENC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						AES-DEC
dupOutHmacKey	HMAC key for duplicated objects	160, 256, 384, 512 - 128 to 256	Symmetric key - CSP	KBKDF		MAC
creSeed	Seed for credential keys derivation	160 to 512 - 128 to 256	Symmetric key - CSP		KAS	KBKDF
creSymKey	Encryption key for credentials	128, 192, 256 - 128 to 256	Symmetric key - CSP	KBKDF		AES-ENC AES-DEC
creHmacKey	HMAC key for credentials	160, 256, 384, 512 - 128 to 256	Symmetric key - CSP	KBKDF		MAC
ephSensEccKey	ECC ephemeral private key	256, 384, 521 - 128 to 256	ECC private key - CSP	KAS-KeyGen		KAS
ephPubEccKey	ECC ephemeral public key	512, 768, 1056 - 128 to 256	ECC public key - PSP	KAS-KeyGen		KAS
ekRsa	Provisioned RSA endorsement key	2048 - 112	RSA private key - CSP	Input during manufacturing		KTS-IFC
ekEcc	Provisioned ECC endorsement key	256, 384 - 128 to 192	ECC private key - CSP	Input during manufacturing		KAS
fuSigECCKey	Field upgrade ECC signature verification key	384 - 192	ECC public key - PSP	Input during manufacturing		SigVer
fuSigLMSKey	Field upgrade LMS signature verification key	32 - 128	LMS public key - PSP	Input during manufacturing		SigVer
seqAuth	Authorization value for hash or HMAC sequence	112 to 512 - 112 to 256	Authentication value / Symmetric key - CSP	N/A		KBKDF MAC
nullSeed	Seed of the null hierarchy	512 - 256	Seed - CSP	ENT-ESV		DRBG
phSeed	Seed of the platform hierarchy	512 - 256	Seed - CSP	ENT-ESV		DRBG
ehSeed	Seed of the endorsement hierarchy	512 - 256	Seed - CSP	ENT-ESV		DRBG
shSeed	Seed of the storage hierarchy	512 - 256	Seed - CSP	ENT-ESV		DRBG
drbgState	Internal state (V and C secret values) of the DRBG (based on SHA256)	256 - 256	State - CSP	DRBG		DRBG
drbgSeed	Seed value for the DRBG	512 - 256	Seed - CSP	ENT-ESV		DRBG
tdrbgState	Internal state (V and C secret values) of the transient DRBG (based on SHA256) used to generate prime numbers for primary RSA keys	256 - 256	State - CSP	DRBG		DRBG
fuSymSeed	Seed used for field upgrade symmetric key derivation	256 - 256	Symmetric key - Neither	Input during manufacturing		KBKDF
fuSymKey	field upgrade symmetric key	256 - 256	Symmetric key - Neither	KBKDF		AES-DEC
diagSymSeed	Seed used for diagnostic symmetric key derivation	256 - 256	Symmetric key - Neither	Input during manufacturing		KBKDF
diagSymKey	diagnostic symmetric key	256 - 256	Symmetric key - Neither	KBKDF		AES-ENC

Table 31: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
nullProof		Static RAM:Plaintext	Until next reset	TPM2_Init	drbgState:Generates contextEncKey:Derived From
phProof		NVRAM:Plaintext	After Use	TPM2_ChangePPS	drbgState:Generates contextEncKey:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
ehProof		NVRAM:Plaintext	After Use	TPM2_ChangeEPS TPM2_Clear	drbgState:Generates contextEncKey:Derived From
shProof		NVRAM:Plaintext	After Use	TPM2_Clear	drbgState:Generates contextEncKey:Derived From
shProofForReseed		NVRAM:Plaintext	After Use	TPM2_Clear	tdrbgState:Reseeded From
platformAuth	Input plaintext to RAM Input protected to RAM	Static RAM:Plaintext	Until next reset	TPM2_Init	sesHmacKey:Derived from, Protects (Integrity) sesSymKey:Encrypts
endorsementAuth	Input plaintext to NVRAM Input protected to NVRAM	NVRAM:Plaintext	After Use	TPM2_Clear TPM2_ChangeEPS	sesHmacKey:Derived from, Protects (Integrity) sesSymKey:Derived from, Protects (Encrypts)
ownerAuth	Input plaintext to NVRAM Input protected to NVRAM	NVRAM:Plaintext	After Use	TPM2_Clear	sesHmacKey:Derived from, Protects (Integrity) sesSymKey:Derived from, Protects (Encrypts)
lockoutAuth	Input plaintext to NVRAM Input protected to NVRAM	NVRAM:Plaintext	After Use	TPM2_Clear	sesHmacKey:Derived from, Protects (Integrity) sesSymKey:Derived from, Protects (Encrypts)
objSeed	Input protected to RAM Input plaintext to RAM Output protected from RAM Output protected from NVRAM	Static RAM:Plaintext NVRAM:Plaintext	Until object zeroization, shift to NVRAM or next reset	TPM2_Init TPM2_Clear TPM2_ChangePPS TPM2_ChangeEPS TPM2_EvictControl TPM2_FlushContext	tdrbgState:Derived From drbgState:Derived From objSymKey:Derived From objHmacKey:Derived From sesHmacKey:Protects (Integrity) sesSymKey:Protects (Encrypts)
objAuth	Input plaintext to RAM Input protected to RAM Output protected from RAM Output protected from NVRAM	Static RAM:Plaintext NVRAM:Plaintext	Until object zeroization, shift to NVRAM or next reset	TPM2_Init TPM2_Clear TPM2_ChangePPS TPM2_ChangeEPS TPM2_EvictControl TPM2_FlushContext	sesHmacKey:Derived from, Protects (Integrity) sesSymKey:Derived from, Encrypts
objSymKey		Dynamic RAM:Plaintext NVRAM:Plaintext	After Use	Automatic	objAuth:Encrypted by objSens:Encrypted by platformAuth:Encrypted by endorsementAuth:Encrypted by ownerAuth:Encrypted by lockoutAuth:Encrypted by
objHmacKey		Dynamic RAM:Encrypted NVRAM:Plaintext	After Use	Automatic	objAuth:Protected by (Integrity) objSens:Protected by (Integrity) platformAuth:Protected by (Integrity) endorsementAuth:Protected by (Integrity) ownerAuth:Protected by (Integrity) lockoutAuth:Protected by (Integrity)
objSens	Input plaintext to RAM Input protected to RAM Output protected from RAM	Static RAM:Plaintext NVRAM:Plaintext	Until object zeroization, shift to NVRAM or next reset	TPM2_Init TPM2_Clear TPM2_ChangePPS TPM2_ChangeEPS	tdrbgState:Generates drbgState:Generates objSeed:Derives objSymKey:Encrypts

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	Output protected from NVRAM			TPM2_EvictControl TPM2_FlushContext	objHmacKey:Protects (Integrity) objPub:Paired With
objPub	Input plaintext to RAM Output plaintext from NVRAM Output plaintext from RAM	Static RAM:Plaintext NVRAM:Plaintext	Until object zeroization, shift to NVRAM or next reset	TPM2_Init TPM2_Clear TPM2_ChangePPS TPM2_ChangeEPS TPM2_EvictControl TPM2_FlushContext	objSens:Paired With
nvAuth	Input plaintext to NVRAM Input protected to NVRAM	NVRAM:Plaintext	After Use	TPM2_NV_UndefineSpace TPM2_NV_UndefineSpaceSpecial	sesHmacKey:Derived from, Protects (Integrity) sesSymKey:Encrypts
sesSalt	Input asym. encrypted to RAM	Dynamic RAM:Plaintext	After Use	Automatic	sesHmacKey:Derived From objPub:Encrypts
sesHmacKey	Input protected to RAM Output protected from RAM	Dynamic RAM:Plaintext	After Use	Automatic	nvAuth:Protected by (Integrity) contextKey:Encrypts contextEncKey:Encrypts platformAuth:Protected by (Integrity) endorsementAuth:Protected by (Integrity) ownerAuth:Protected by (Integrity) lockoutAuth:Protected by (Integrity) objAuth:Protected by (Integrity) seqAuth:Derives; Protected by (Integrity) dupInSymKey:Protected by (Integrity)
sesSymKey		Dynamic RAM:Plaintext	After Use	Automatic	sesHmacKey:Derives platformAuth:Derives; Encrypts endorsementAuth:Derives; Encrypts ownerAuth:Derives; Encrypts lockoutAuth:Derives; Encrypts objAuth:Derives; Encrypts seqAuth:Derives; Encrypts nvAuth:Derives; Encrypts dupInSymKey:Encrypted by
contextKey		Static RAM:Plaintext	Until next reset	TPM2_Init	drbgState:Generates contextEncKey:Derived From
contextEncKey		Dynamic RAM:Plaintext	After Use	Automatic	contextKey:Derives nullProof:Derives phProof:Derives ehProof:Derives shProof:Derives
dupInSymKey	Input plaintext to RAM Input protected to RAM Output plaintext from RAM Output protected from RAM	Dynamic RAM:Plaintext	After Use	Automatic	sesSymKey:Encrypts sesHmacKey:Protects (Integrity) objSens:Encrypted by

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
dupSeed	Input asym. encrypted to RAM Output asym. encrypted to RAM	Dynamic RAM:Plaintext	After Use	Automatic	objPub:Encrypts dupOutSymKey:Derived from dupOutHmacKey:Derived from
dupOutSymKey		Dynamic RAM:Plaintext	After Use	Automatic	dupSeed:Derives objSens:Encrypted by objAuth:Encrypted by objSeed:Encrypted by
dupOutHmacKey		Dynamic RAM:Plaintext	After Use	Automatic	dupSeed:Derives objSens:Protects (Integrity) objAuth:Protects (Integrity) objSeed:Protects (Integrity)
creSeed	Input asym. encrypted to RAM Output asym. encrypted to RAM	Dynamic RAM:Plaintext	After Use	Automatic	creSymKey:Derives creHmacKey:Derives objPub:Encrypts
creSymKey		Dynamic RAM:Plaintext	After Use	Automatic	creSeed:Derived From
creHmacKey		Dynamic RAM:Plaintext	After Use	Automatic	creSeed:Derived From
ephSensEccKey		Dynamic RAM:Plaintext	After Use	Automatic	drbgState:Generates
ephPubEccKey	Input plaintext to RAM Output plaintext from RAM	Dynamic RAM:Plaintext	After Use	Automatic	ephSensEccKey:Derives
ekRsa		NVRAM:Plaintext	After Use	TPM2_VendorCmdZeroizeEK	objSens:Derived From
ekEcc		NVRAM:Plaintext	After Use	TPM2_VendorCmdZeroizeEK	objSens:Derived From
fuSigECCKey		NVRAM:Plaintext	After Use	N/A	
fuSigLMSKey		NVRAM:Plaintext	After Use	N/A	
seqAuth	Input plaintext to RAM Input protected to RAM Output protected from RAM	Dynamic RAM:Plaintext	Until use of zeroization command or next reset	TPM2_SequenceComplete TPM2_EventSequenceComplete	sesSymKey:Derived From sesHmacKey:Derived From
nullSeed		Static RAM:Plaintext	Until next reset	TPM2_Init	tdrbgState:Instantiated with
phSeed		NVRAM:Plaintext	After Use	TPM2_ChangePPS	tdrbgState:Instantiated with
ehSeed		NVRAM:Plaintext	After Use	TPM2_ChangeEPS	tdrbgState:Instantiated with
shSeed		NVRAM:Plaintext	After Use	TPM2_Clear	tdrbgState:Instantiated with
drbgState		Static RAM:Plaintext	Until next reset or use of TPM2_Clear	TPM2_Init TPM2_Clear	drbgSeed:Instantiates
drbgSeed		Dynamic RAM:Plaintext	After Use	Automatic	drbgState:Instantiated with
tdrbgState		Dynamic RAM:Plaintext	After Use	Automatic	nullSeed:Instantiates phSeed:Instantiates ehSeed:Instantiates shSeed:Instantiates
fuSymSeed		NVRAM:Plaintext	After Use	N/A	fuSymKey:Derived From
fuSymKey		Dynamic RAM:Plaintext	After Use	Automatic	fuSymSeed:Derives

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
diagSymSeed		NVRAM:Plaintext	After Use	N/A	diagSymKey:Derived From
diagSymKey		Dynamic RAM:Plaintext	After Use	Automatic	diagSymSeed:Derives

Table 32: SSP Table 2

Name	Description	Size (bits)	Strength	Type	Generated by	Established by	Used by	Inputs / Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
nullProof	Proof (secret value) of the null hierarchy	512	256	Symmetric key	DRBG	N/A	KBKDF MAC	-	Static RAM	Until next reset	TPM2_Init	CSP	Derived from drbgState contextEncKey can be derived from nullProof
phProof	Proof (secret value) of the platform hierarchy	512	256	Symmetric key	DRBG	N/A	MAC	-	NVRAM	After use	TPM2_ChangePPS	CSP	Derived from drbgState contextEncKey can be derived from phProof
ehProof	Proof (secret value) of the endorsement hierarchy	512	256	Symmetric key	DRBG	N/A	MAC	-	NVRAM	After use	TPM2_ChangeEPS TPM2_Clear	CSP	Derived from drbgState contextEncKey can be derived from ehProof
shProof	Proof (secret value) of the storage hierarchy	512	256	Symmetric key	DRBG	N/A	KBKDF MAC	-	NVRAM	After use	TPM2_Clear	CSP	Derived from drbgState contextEncKey can be derived from shProof
shProofForReseed	Random value	512	256	Entropy source	ENT-ESV	N/A	DRBG	-	NVRAM	After use	TPM2_Clear	CSP	tdrbgState is reseeded with shProofForReseed
platformAuth	Authentication value for the platform hierarchy	512	128 to 256 (depending on the underlying hash algorithm used)	Authentication value / Symmetric key	N/A	N/A	KBKDF MAC	Input protected to RAM or Input plaintext to RAM	Static RAM	Until next reset	TPM2_Init	CSP	sesHmacKey can be derived from platformAuth New input platformAuth value can be wrapped by sesSymKey and integrity protected by sesHmacKey

Name	Description	Size (bits)	Strength	Type	Generated by	Established by	Used by	Inputs / Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
endorsementAuth	Authentication value for the endorsement hierarchy	512	128 to 256 (depending on the underlying hash algorithm used)	Authentication value / Symmetric key	N/A	N/A	KBKDF MAC	Input protected to NVRAM or Input plaintext to NVRAM	NVRAM	After use	TPM2_Clear TPM2_ChangeEPS	CSP	sesHmacKey and sesSymKey can be derived from endorsementAuth New input endorsementAuth value can be wrapped by sesSymKey and integrity protected by sesHmacKey
ownerAuth	Authentication value for the storage hierarchy	512	128 to 256 (depending on the underlying hash algorithm used)	Authentication value / Symmetric key	N/A	N/A	KBKDF MAC	Input protected to NVRAM or Input plaintext to NVRAM	NVRAM	After use	TPM2_Clear	CSP	sesHmacKey and sesSymKey can be derived from ownerAuth New input ownerAuth value can be wrapped by sesSymKey and integrity protected by sesHmacKey
lockoutAuth	Authentication value for the lockout hierarchy	512	128 to 256 (depending on the underlying hash algorithm used)	Authentication value / Symmetric key	N/A	N/A	KBKDF MAC	Input protected to NVRAM or Input plaintext to NVRAM	NVRAM	After use	TPM2_Clear	CSP	sesHmacKey and sesSymKey can be derived from lockoutAuth New input lockoutAuth value can be wrapped by sesSymKey and integrity protected by sesHmacKey
nullSeed	Seed of the null hierarchy	512	256	Seed	ENT-ESV	N/A	DRBG	-	Static RAM	Until next reset	TPM2_Init	CSP	tdrbgState can be instantiated by nullSeed
phSeed	Seed of the platform hierarchy	512	256	Seed	ENT-ESV	N/A	DRBG	-	NVRAM	After use	TPM2_ChangePPS	CSP	tdrbgState can be instantiated by phSeed

Name	Description	Size (bits)	Strength	Type	Generated by	Established by	Used by	Inputs / Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
ehSeed	Seed of the endorsement hierarchy	512	256	Seed	ENT-ESV	N/A	DRBG	-	NVRAM	After use	TPM2_ChangeEPS	CSP	tdrbgState can be instantiated by ehSeed
shSeed	Seed of the storage hierarchy	512	256	Seed	ENT-ESV	N/A	DRBG	-	NVRAM	After use	TPM2_Clear	CSP	tdrbgState can be instantiated by shSeed
objSeed	Seed value for object generation	512	128 to 256	Data, Symmetric key	DRBG KBKDF	N/A	SHA KBKDF	Input protected to RAM Input plaintext to RAM Output protected from RAM Output protected from NVRAM	Static RAM NVRAM	Until object zeroization, shift to NVRAM or next reset	TPM2_Clear TPM2_ChangePPS TPM2_ChangeEPS TPM2_EvictControl TPM2_FlushContext TPM2_Init	CSP	can be derived from tdrbgState for primary objects, from drbgState for ordinary objects can be protected by sesHmacKey and sesSymKey objSymKey and objHmacKey are derived from objSeed
objAuth	Object's authorization value	112 to 512	112 to 256	Authentication value / Symmetric key	N/A	N/A	MAC KBKDF	Input protected to RAM Input plaintext to RAM Output protected from RAM Output protected from NVRAM	Static RAM NVRAM	Until object zeroization, shift to NVRAM or next reset	TPM2_Clear TPM2_ChangePPS TPM2_ChangeEPS TPM2_EvictControl TPM2_FlushContext TPM2_Init	CSP	can be protected by sesHmacKey and sesSymKey sesHmacKey and sesSymKey can be derived from objAuth
objSymKey	Encryption key of object private part	256	256	Symmetric key	KBKDF	N/A	AES-ENC, AES-DEC	-	Dynamic RAM NVRAM	After use	Automatic	CSP	can wrap platformAuth / endorsementAuth / ownerAuth / lockoutAuth / objAuth/objSens
objHmacKey	Integrity key of object private part	160, 256, 384, 512	128 to 256	Symmetric key	KBKDF	N/A	MAC	-	Dynamic RAM NVRAM	After use	Automatic	CSP	can protect platformAuth / endorsementAuth / ownerAuth / lockoutAuth / objAuth/objSens

Name	Description	Size (bits)	Strength	Type	Generated by	Established by	Used by	Inputs / Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
objSens	Object private part	2048, 3072, 4096 (RSA) 128, 192, 256 (AES) 256, 384, 512, 521, (ECC) 112 to 1024 (HMAC)	112 to 256	Symmetric or asymmetric private key	CKG KBKDF KAS-KeyGen KeyGen	N/A	AES-ENC, AES-DEC, KBKDF MAC KAS SigGen (RSA, ECDSA),	Input protected to RAM Input plaintext to RAM Output protected from RAM Output protected from NVRAM	Static RAM NVRAM	Until object zeroization, shift to NVRAM or next reset	TPM2_Clear TPM2_ChangePPS TPM2_ChangeEPS TPM2_EvictControl TPM2_FlushContext TPM2_Init	CSP	can be generated from tdrbgState for primary objects, from drbgState for ordinary objects and derived from objSeed of its parent for derived objects objSymKey encrypts objSens objHmacKey can integrity protect objSens objPub: Paired With
objPub	Object public part	2048, 3072, 4096 (RSA) 2*256, 2*384, 2*521 (ECC)	112 to 256	Asymmetric public key	KeyGen (ECDSA, RSA) KAS-KeyGen	N/A	KAS, KTS-IFC, SigVer (RSA, ECDSA),	Input plaintext to RAM Output plaintext from RAM Output plaintext from NVRAM	Static RAM NVRAM	Until object zeroization, shift to NVRAM or next reset	TPM2_Clear TPM2_ChangePPS TPM2_ChangeEPS TPM2_EvictControl TPM2_FlushContext TPM2_Init	PSP	objSens: Paired With
nvAuth	Authorization of NV index	112 to 512	112 to 256	Authentication value / Symmetric key	N/A	N/A	KBKDF MAC	Input protected to NVRAM Input plaintext to NVRAM	NVRAM	After use	TPM2_NV_UndefineSpace TPM2_NV_UndefineSpaceSpecial	CSP	sesHmacKey can be derived from nvAuth New input nvAuth value can be wrapped by sesSymKey and integrity protected by sesHmacKey
sesSalt	Salt for keys diversification	160, 256, 384, 512	128 to 256	Symmetric key	N/A	KAS	KBKDF	Input asym. encrypted to RAM	Dynamic RAM	After use	Automatic	CSP	sesHmacKey is derived from sesSalt

Name	Description	Size (bits)	Strength	Type	Generated by	Established by	Used by	Inputs / Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
													objPub wraps sesSalt
sesHmacKey	HMAC session key	160, 256, 384, 512	128 to 256	Symmetric key	KBKDF	N/A	KBKDF MAC	Input protected to RAM Output protected from RAM	Dynamic RAM	After use	Automatic	CSP	protects nvAuth / platformAuth / endorsementAuth / ownerAuth / lockoutAuth / objAuth / seqAuth / dupInSymKey derived from seqAuth contextKey and contextEncKey keys can wrap sesHmacKey
sesSymKey	Encrypted session key	128, 192, 256	128 to 256	Symmetric key	KBKDF	N/A	AES-ENC, AES-DEC	-	Dynamic RAM	After use	Automatic	CSP	derived from and encrypts sesHmacKey and platformAuth / endorsementAuth / ownerAuth / lockoutAuth / objAuth / seqAuth
contextKey	Derivation key for context protection	128	128	Symmetric key	DRBG	N/A	KBKDF	-	Static RAM	Until next reset	TPM2_Init	CSP	generated from drbgState contextEncKey is derived from contextKey
contextEncKey	Wrapping key for context protection	256	256	Symmetric key	KBKDF	N/A	AES-ENC, AES-DEC	-	Dynamic RAM	After use	Automatic	CSP	derived from contextKey and nullProof / phProof / ehProof / shProof
dupInSymKey	Wrapping key for duplicated object	128, 192, 256	128 to 256	Symmetric key	DRBG	N/A	AES-ENC, AES-DEC	Input plaintext to RAM Input protected to RAM Output plaintext from RAM Output protected from RAM	Dynamic RAM	After use	Automatic	CSP	can be wrapped by sesSymKey and protected by sesHmacKey Encrypts objSens

Name	Description	Size (bits)	Strength	Type	Generated by	Established by	Used by	Inputs / Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
dupSeed	Seed for protection keys derivation	160 to 512	128 to 256	Symmetric key	DRBG KAS	KAS	KBKDF	Input asym. encrypted to RAM Output asym. encrypted from RAM	Dynamic RAM	After use	Automatic	CSP	encrypted by objPub key (KTS-IFC or KAS) dupOutSymKey and dupOutHmacKey are derived from dupSeed
dupOutSymKey	Encryption key for duplicated objects	128, 192, 256	128 to 256	Symmetric key	KBKDF	N/A	AES-ENC, AES-DEC	-	Dynamic RAM	After use	Automatic	CSP	derived from dupSeed wraps objSens, objAuth, objSeed
dupOutHmacKey	MAC key for duplicated objects	160, 256, 384, 512	128 to 256	Symmetric key	KBKDF	N/A	MAC	-	Dynamic RAM	After use	Automatic	CSP	derived from dupSeed protects objSens, objAuth, objSeed
creSeed	Seed for credential keys derivation	160 to 512	128 to 256	Symmetric key	N/A	KAS	KBKDF	Input asym. encrypted to RAM Output asym. encrypted from RAM	Dynamic RAM	After use	Automatic	CSP	creSymKey and creHmacKey are derived from creSeed Encrypted by objPub
creSymKey	Encryption key for credentials	128, 192, 256	128 to 256	Symmetric key	KBKDF	N/A	AES-ENC, AES-DEC	-	Dynamic RAM	After use	Automatic	CSP	derived from creSeed
creHmacKey	HMAC key for credentials	160, 256, 384, 512	128 to 256	Symmetric key	KBKDF	N/A	MAC	-	Dynamic RAM	After use	Automatic	CSP	derived from creSeed
ephSensEccKey	ECC ephemeral private key	256, 384, 521	128 to 256	ECC private key	KAS-KEYGEN	N/A	KAS	-	Dynamic RAM	After use	Automatic	CSP	derived from drbgState
ephPubEccKey	ECC ephemeral public key	512, 768, 1056	128 to 256	ECC public key	KAS-KEYGEN	N/A	KAS	Input plaintext to RAM Output plaintext from RAM	Dynamic RAM	After use	Automatic	PSP	generated from ephSensEccKey
ekRsa	Provisioned RSA endorsement key	2048	112	RSA private key	Other – Input during manufacturing	N/A	KTS-IFC	-	NVRAM	After use	TPM2_VendorCmdZeroizeEK	CSP	objSens is generated from ekRsa

Name	Description	Size (bits)	Strength	Type	Generated by	Established by	Used by	Inputs / Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
ekEcc	Provisioned ECC endorsement key	256, 384	128 to 192	ECC private key	Other – Input during manufacturing	N/A	KAS	-	NVRAM	After use	TPM2_VendorCmdZeroizeEK	CSP	objSens is generated from ekEcc
fuSigECCKey	Field upgrade ECC signature verification key	384	192	ECC public key	Other – Input during manufacturing	N/A	SigVer (ECDSA)	-	NVRAM	After use	-	PSP	-
fuSigLMSKey	Field upgrade LMS signature verification key	32	128	LMS public key	Other – Input during manufacturing	N/A	SigVer (LMS)	-	NVRAM	After use	-	PSP	-
seqAuth	Authorization value for hash or HMAC sequence	112 to 512	112 to 256	Authentication value / Symmetric key	N/A	N/A	KBKDF MAC	Input plaintext to RAM Input protected to RAM Output protected from RAM	Dynamic RAM	Until use of zeroization command or next reset	TPM2_SequenceComplete TPM2_EventSequenceComplete	CSP	sesSymKey and sesHmacKey are derived from seqAuth
drbgState	Internal state (V and C secret values) of the DRBG (based on SHA256)	256	256	State	DRBG	N/A	DRBG	-	Static RAM	Until next reset or use of TPM2_Clear	TPM2_Clear TPM2_Init	CSP	seeded by drbgSeed
drbgSeed	Seed value for the DRBG	512	256	Seed	ENT-ESV	N/A	DRBG	-	Dynamic RAM	After use	Automatic	CSP	seeds drbgState
tdrbgState	Internal state (V and C secret values) of the transient DRBG (based on SHA256) used to generate prime numbers for primary RSA keys.	256	256	State	DRBG	N/A	DRBG	-	Dynamic RAM	After use	Automatic	CSP	instantiated by nullSeed / phSeed / ehSeed / shSeed
fuSymSeed	Seed used for field upgrade symmetric key derivation	256	256	Symmetric key	Other – Input during Manufacturing	N/A	KBKDF	-	NVRAM	After use	None	non-SSP	fuSymKey is derived from fuSymSeed
fuSymKey	field upgrade symmetric key	256	256	Symmetric key	KBKDF	N/A	AES-DEC	-	Dynamic RAM	After use	Automatic	non-SSP	fuSymKey is derived from fuSymSeed

Name	Description	Size (bits)	Strength	Type	Generated by	Established by	Used by	Inputs / Outputs	Storage	Temporary Storage Duration	Zeroization	Category	Related SSPs
diagSymSeed	Seed used for diagnostic symmetric key derivation	256	256	Symmetric key	Other – Input during Manufacturing	N/A	KBKDF	-	NVRAM	After use	None	non-SSP	diagSymKey is derived from diagSymSeed
diagSymKey	diagnostic symmetric key	256	256	Symmetric key	KBKDF	N/A	AES-ENC	-	Dynamic RAM	After use	Automatic	non-SSP	diagSymKey is derived from diagSymSeed

Next table gives the security strength of a key depending on the underlying algorithm used and its size:

Algorithm*	Underlying algorithm	Key size (bits)	Security strength (bits)
KBKDF	SHA-1	size ≥ 128	128
		size < 128	Key size
	SHA2-256	size ≥ 192	192
		size < 192	Key size
	SHA2-384 SHA2-512	size ≥ 256	256
		size < 256	Key size
HMAC	SHA-1	size ≥ 128	128
		size < 128	Key size
	SHA2-256	size ≥ 192	192
		size < 192	Key size
	SHA2-384 SHA2-512	size ≥ 256	256
		size < 256	Key size
DRBG	SHA2-256	-	256
AES	-	128 / 192 / 256	128 / 192 / 256
RSA	-	2048 / 3072 / 4096	112 / 128 / 142
ECC	-	256 / 384 / 521	128 / 192 / 256

Table 33 – Security Strength of a Key Depending on the Underlying Algorithm Used and its Size

9.5 Transitions

The module only supports the use of SHA-1 in the Approved mode of operation for non-digital signature applications as permitted by SP800-131Ar2. The module only permits the use of SHA-1 for the purposes of digital signatures in the non-Approved mode.

9.6 Additional Information

N/A

10 Self-Tests

10.1 Pre-Operational Self-Tests

The Module performs self-tests to ensure the proper operation of the Module. Per FIPS 140-3 these are categorized as either pre-operational self-tests or conditional self-tests. Pre-operational self–tests are available on demand by power cycling the Module. The Module performs the following pre-operational self-tests in the table below:

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Firmware integrity test	CRC 16	EDC	SW/FW Integrity	Successful execution of TPM2_Startup command indicates tests have been run	FW integrity is verified by computing an EDC (CRC-16 [ISO13239]) and comparing it to reference values.
HW integrity	HW registers verification	KAT	Critical Function	Successful execution of TPM2_Startup command indicates tests have been run	HW integrity is guaranteed via check of HW sensors. If failure is detected during boot sequence, status is set to FAIL, and error is returned.

Table 34: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

The Module performs the following conditional self-tests in the table below. The bit index indicated in the “Indicator” column corresponds to the index in the algos_status field in the TPM2_GetTestResult response.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ENC (A5356)	AES-128-CBC	KAT	CAST	Bit #7 clear	AES CBC 128 encryption of known data compared to a reference value.	Power On
AES-DEC (A5356)	AES-128-CBC	KAT	CAST	Bit #7 clear	AES CBC 128 decryption of known encrypted data and comparison to the expected plaintext data	Power On
ECDSA KeyGen (FIPS186-4) (A5358)	P-256, P-384, P-521	PCT	PCT	Key creation failure	Depending on the key purpose (signing or key establishment) an ECDSA signature is generated (k fixed and the message varies) and verified with pairwise consistency test as defined by [56Ar3] or a scalar multiplication is done and compared to the public key.	Upon ECC Key Generation
ECDSA SigGen (FIPS186-4) (A5358)	NIST P-256	KAT	CAST	Bit #10 clear	ECDSA signature generation on known data with known key and k. Output of signature is compared to a reference signature.	Power On
ECDSA SigVer (FIPS186-4) (A5358)	NIST P-256	KAT	CAST	Bit #10 clear	ECDSA signature verification on known signature with known key and k.	Power On
Entropy	RCT and APT	[90B] Health-Test	CAST	Bit #1 clear	AIS31 and [90B] (RCT and APT) start-up health tests on ESV #E41 output sequence. If test fails, test status is set to FAIL, and error is returned	At each random bits generation
Firmware loading	ECDSA P-384 and LMS	Signature Verification	SW/FW Load	Error returned on FW loading command	Verification of chained digest and signature to ensure authentication of the FW	Upon firmware load
Hash DRBG (A5351)	SHA2-256	KAT	CAST	Bit #1 clear	Instantiate then Reseed are seeded with a known seed value (64 bytes). Random is then generated with Generate API to output a 32-bytes value compared to a reference value (single test sequence done in accordance with §11.3 of [90A])	Power On
HMAC-SHA-1 (A5355)	HMAC-SHA1	KAT	CAST	Bit #5 clear	HMAC on known data and known key. Comparison of output to an expected MAC value (20 bytes)	Power On
KAS-ECC Sp800-56Ar3 (A5358)	NIST P-256	KAT	CAST	Bit #9 clear	Primitive "Z" Computation and key derivation are implemented: a known private key d is used with a known point P of NIST P-256 curve to compute Q = dP. Key derivation	Power On

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
					of Q performed with SHA-1 underlying algorithm to output a key of 20 bytes that is compared to a refence value	
KDF SP800-108 (A5354)	N/A	KAT	CAST	Bit #6 clear	KDF on known data and known label. Comparison of output to an expected derivation value (32 bytes)	Power On
LMS SigVer (A5360)	LMOTS_SHA256_N32_W4 LMS_SHA256_M32_H10	KAT	CAST	Bit #8 clear	LMS signature verification of known signature with known data and known key.	Power On
RSA SigGen (FIPS186-5) (A5357)	RSASSA-PKCS1-v1_5	KAT	CAST	Bit #12 clear	RSA signature generation on known data with a known key. Output of signature is compared to a reference signature (covers also KTS-IFC functionality)	Power On
RSA SigVer (FIPS186-5) (A5357)	RSASSA-PKCS1-v1_5	KAT	CAST	Bit #12 clear	RSA signature verification on a known signature with a known key (covers also KTS-IFC functionality)	Power On
RSA KeyGen (FIPS186-5) (A5357)	2048, 3072 or 4096-bit	PCT	PCT	Key creation failure	Depending on the key purpose (signing or encrypting) indicated in sign attribute of the key, encryption/decryption or signing/verification is done on known data	Upon RSA Key Generation
SHS	SHA1, SHA2-256, SHA2-512, SHA3-256	KAT	CAST	Bit #2 clear Bit #3 clear Bit #4 clear	Hash of known data and comparison of output to an expected digest. SHA-1, SHA2-256, SHA2-512 are tested twice to cover each of the two implementations covered by CAVP Cert. #A5352 and #A5353.	Power On

Table 35: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Firmware integrity test	EDC	SW/FW Integrity	On demand	Manually
HW integrity	KAT	Critical Function	On demand	Manually

Table 36: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ENC (A5356)	KAT	CAST	On Demand	Manually
AES-DEC (A5356)	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A5358)	PCT	PCT	N/A	Manually
ECDSA SigGen (FIPS186-4) (A5358)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A5358)	KAT	CAST	On Demand	Manually
Entropy	[90B] Health-Test	CAST	On Demand	Manually
Firmware loading	Signature Verification	SW/FW Load	On Demand	Manually
Hash DRBG (A5351)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A5355)	KAT	CAST	On Demand	Manually
KAS-ECC Sp800-56Ar3 (A5358)	KAT	CAST	On Demand	Manually
KDF SP800-108 (A5354)	KAT	CAST	On Demand	Manually
LMS SigVer (A5360)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5357)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5357)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA KeyGen (FIPS186-5) (A5357)	PCT	PCT	N/A	Manually
SHS	KAT	CAST	On Demand	Manually

Table 37: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
ES1	The Module fails a KAT, PCT, FW or HW integrity verification, [90B] health test	The Module enters the failure state	Reboot/Power cycle the module	Outputs return code of TPM_RC_FAILURE, otherwise it indicates successful completion by TPM_RC_SUCCESS
ES2	The Module fails a firmware loading test	The Module returns to normal state	None	Return code different from TPM_RC_SUCCESS sent on firmware upgrade start command

Table 38: Error States

All cryptographic functions are inhibited while the module is in an error state. Successful completion of self-tests can be verified through use of TPM2_GetTestResult command. The first 4 bytes of response indicate self-tests status. If they are equal to 0, self-tests completed successfully. If not, the subsequent 4 bytes indicate the list of algorithms not fully self-tested.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

Installation and Initialization:

The following steps must be performed in order to securely install, initialize, and start up the Module in the FIPS 140-3 Approved mode of operation:

- Connection of the Module with its environment must be done according to the pinout description detailed in section 3.2.
- Command TPM2_SetCapability with the following parameters must be sent to the Module to configure it in FIPS 140-3 level2 mode:
  - setCapability = TPM_CAP_N_CONFIGS (0x80100000)
  - configuration = TPM_PT_CONFIG_FIPS_SL2_MODE (0x00000000)
  - enable = YES (0x01)
  - lock = YES (0x01)

Module operation:

Once configured in FIPS 140-3 level2, the following restrictions are enforced by the Module:

- The default authValue of hierarchies (owner, endorsement, lockout, platform) must be changed prior being used in an authorization session. If not done, the authorization will be reported as failed.
- Use of other authorization sessions than the ones described in section 4.1 is prohibited except for the following use cases:
  - First use or use after TPM2_Clear (for the owner, endorsement, or lockout authValue) or use after reset (for the platform authValue) of the TPM2_HierarchyChangeAuth command to change the default authValue of the hierarchies.
- If the minimum length of an object's authValue is less than 112 bits (14 bytes), the creation of the object will fail with the error TPM_RC_AUTHSIZE. This concerns keys, NV indices and Hash/HMAC sessions.
- Use of a policy authorization session will fail and report TPM_RC_AUTH_FAIL if one of the following policy commands is not part of the policy:

Policy commands*	Authentication mechanism	Description
TPM2_PolicyAuthValue	Message Authentication Code	*authValue* of authorized entity is used as HMAC key in authorization HMAC (as for HMAC session)
TPM2_PolicySigned	Public Key Digital Signature Algorithm or Message Authentication Code	Signature with asymmetric or HMAC key
TPM2_PolicyAuthorize	Message Authentication Code	Signature with HMAC key being one of the hierarchy proofs
TPM2_PolicySecret	Message Authentication Code	*authValue* of reference entity is provided in HMAC session, or policy session containing TPM2_PolicyAuthValue

TPM2_PolicyTicket	Message Authentication Code	Signature with HMAC key (one of the proofs) generated by TPM2_PolicySigned or TPM2_PolicySecret
Bound session	Message Authentication Code	*authValue* of bound entity is used as KDK generated from KBKDF in session key derivation

Table 39 – List of policy commands to use in a policy session

TPM is operated in an approved mode of operation as long as no non-approved service using a non-approved algorithm is used. No specific rules of operation are required to operate this module at FIPS 140-3 Level 2.

TPM is in normal operation mode when all pre-operational and conditional self-tests (apart from firmware load and PCT tests) are complete. All approved and non-approved services with the corresponding indicator reporting if the service uses an approved cryptographic algorithm or a security function.

11.2 Administrator Guidance

No specific initialization procedure is required.

11.3 Non-Administrator Guidance

No initialization procedures are required.

11.4 Design and Rules

Rules of Operation

1. The Module provides two operator roles: the Cryptographic Officer and the User role. Each role is associated with a set of services as detailed in Section 4.3.
2. The Module, evaluated at FIPS 140-3 Level 2, requires authentication to access some of the services as detailed in Section 4.1.
3. The Module allows the operator to initiate power-up self-tests by power cycling or resetting the Module.
4. Power up self-tests do not require any operator action.
5. Data output is inhibited during key generation, self-tests, zeroisation, firmware loading, and error states.
6. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the Module.
7. The Module does not support concurrent operators.
8. The Module does not support a maintenance interface or role.
9. The Module does not support manual key entry method.

10. The Module does not have any proprietary external input/output devices used for entry/output of data.
11. The Module does not output intermediate key values.
12. The Module does not provide bypass services or ports/interfaces.
13. The Module does not support a self-initiated cryptographic output capability.
14. For all zeroisation methods, the module must be in direct control of the operator.

11.5 Maintenance Requirements

N/A

11.6 End of Life

End-of-life of the product requires the following zeroisation commands to be executed to remove all CSPs from the memory of the module:

- TPM2_Init
- TPM2_Clear
- TPM2_ChangeEPS
- TPM2_ChangePPS
- TPM2_VendorCmdZeroizeEK
- TPM2_NV_UndefineSpace or TPM2_NV_UndefineSpaceSpecial

12 Mitigation of Other Attacks

The Module does not implement any mitigation method against other attacks.

References and Definitions

The following standards are referred to in this Security Policy.

Abbreviation*	Full Specification Name
[TPM2.0 Part1]	*TPM2.0 Main, Part 1, Architecture, rev 1.59, TCG*
[TPM2.0 Part2]	*TPM2.0 Main, Part 2, Structures, rev 1.59, TCG*
[TPM2.0 Part3]	*TPM2.0 Main, Part 3, Commands, rev 1.59, TCG*
[TPM2.0 Part4]	*TPM2.0 Main, Part 4, Supporting routines, rev 1.59, TCG*
[PTP 1.06]	*TCG PC Client Platform TPM Profile (PTP) Specification, rev. 1.06*
[FIPS TCG]	*TCG FIPS 140-3 guidance for TPM 2.0, version 1.0, rev. 1, January 30, 2024*
[ISO19790]	*International Standard, ISO/IEC 19790, Information technology — Security techniques — Test requirements for cryptographic modules, Third edition, March 2017*
[ISO24759]	*International Standard, ISO/IEC 24759, Information technology — Security techniques — Test requirements for cryptographic modules, Second and Corrected version, 15 December 2015*
[ISO13239]	*International Standard, ISO/IEC 13239, Information technology — Telecommunications and information exchange between systems — High-level data link control (HDLC) procedures, July 2002*
[140-3]	*Security Requirements for Cryptographic Modules, March 22, 2019*
[140]	*NIST Special Publication 800-140, FIPS 140-3 Derived Test Requirements (DTR), CMVP Validation Authority Updates to ISO/IEC 24759, March 2020*
[140A]	*NIST Special Publication 800-140A, CMVP Documentation Requirements, CMVP Validation Authority Updates to ISO/IEC 24759, March 2020*
[140Br1]	*NIST Special Publication 800-140B revision 1, CMVP Security Policy Requirements, CMVP Validation Authority Updates to ISO/IEC 24759 and ISO/IEC 19790 Annex B, November 2023*
[140C]	*NIST Special Publication 800-140Cr2, CMVP Approved Security Functions, CMVP Validation Authority Updates to ISO/IEC 24759, July 2023*
[140D]	*NIST Special Publication 800-140Dr2, CMVP Approved Sensitive Security Parameter Generation and Establishment Methods, CMVP Validation Authority Updates to ISO/IEC 24759, July 2023*
[140E]	*NIST Special Publication 800-140E, CMVP Approved Authentication Mechanisms, CMVP Validation Authority Requirements for ISO/IEC 19790:2012 Annex E and ISO/IEC 24759 Section 6.17, March 2020*
[140F]	*NIST Special Publication 800-140Fr1, CMVP Approved Non-Invasive Attack Mitigation Test Metrics, CMVP Validation Authority Updates to ISO/IEC 24759, August 2021*
[IG]	*Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program, October 23, 2024*
[108]	*NIST Special Publication 800-108r1-upd1, Recommendation for Key Derivation Using Pseudorandom Functions (Revised), August 2022*
[131A]	*Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, Revision 2, March 2019*

Abbreviation*	Full Specification Name
[133]	*NIST Special Publication 800-133, Recommendation for Cryptographic Key Generation, Revision 2, June 2020*
[135]	*National Institute of Standards and Technology, Recommendation for Existing Application-Specific Key Derivation Functions, Special Publication 800-135rev1, December 2011*
[186]	*National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-5, Feb 2023*
[197]	*National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197-upd1, May, 2023*
[198]	*National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198-1, July, 2008*
[180]	*National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication 180-4, August 2015*
[202]	*FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, FIPS PUB 202, August 2015*
[208]	*National Institute of Standards and Technology, Recommendation for Stateful Hash-Based Signature Schemes, October 2020*
[38A]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, Special Publication 800-38A, December 2001*
[56Ar3]	*NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, April 2018*
[56Br2]	*NIST Special Publication 800-56B Revision 2, Recommendation for Pair-Wise Key Establishment Schemes Using Finite Field Cryptography, March 2019*
[90A]	*National Institute of Standards and Technology, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Special Publication 800-90A, Revision 1, June 2015*
[90B]	*National Institute of Standards and Technology, Recommendation for the Entropy Sources Used for Random Bit Generation, Special Publication 800-90B, January 2018*

Table 40 – References

Acronym*	Definition
APT	Adaptive Proportion Test
BN P-256	Barreto-Naehrig 256-bit elliptic curve
FW	Firmware
HW	Hardware
KAT	Know Answer Test
I^2C	Inter-Integrated Circuit
MPU	Memory Protection Unit
RCT	Repetition Count Test
SPI	Serial Peripheral Interface

Acronym*	Definition
SSP	Sensitive Security Parameter
TCG	Trusted Computing Group
TPM	Trusted Platform Module

Table 41 – Acronyms and Definitions