Thales Luna K7 Cryptographic Module

LEVEL 3 NON-PROPRIETARY SECURITY POLICY

002-010935-003
Rev. U
January 31, 2025

Document Information

Document Part Number	002-010935-003
Initial Release Date	February 23, 2024
Update Release Date	January 31, 2025

Trademarks, Copyrights, and Third-Party Software

Disclaimer

All information herein is either public information or is the property of and owned solely by Thales and/or its subsidiaries who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Thales's information.

This document can be copied or distributed for informational, non-commercial, internal and personal use only provided that:

- The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any network computer or broadcast in any media other than on the NIST CMVP validation list and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales makes no warranty as to the value or accuracy of information contained herein.

Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

CONTENTS

ACRONYMS AND ABBREVIATIONS 6

REFERENCES 11

PREFACE 14

1 General 15
- 1.1 Security Level 15

2 Cryptographic Module Specification 16
- 2.1 Module Overview 16
- 2.2 Module Description 16
- 2.3 Test Configuration 20
- 2.4 Approved Algorithms 21
- 2.5 Non-Approved Algorithms 39

3 Cryptographic Module Interfaces 42
- 3.1 Ports and Interface Overview 42

4 Roles, Services, and Authentication 46
- 4.1 Roles 46
- 4.2 Roles and Authentication 52
  - 4.2.1 Authentication Mechanism Summary 52
  - 4.2.2 Activation 53
  - 4.2.3 Account lockout behaviors 53
- 4.3 Approved Services 53
- 4.4 Non-Approved Services 86

5 Software/Firmware Security 91
- 5.1 Firmware Integrity 91
- 5.2 FM Integrity 91
- 5.3 Firmware Load 91
- 5.4 Firmware Components 91

6 Operational Environment 93

7 Physical Security 94
- 7.1 Mechanism Summary 94
  - 7.1.1 Module Construction 94
  - 7.1.2 External Event trigger 94
  - 7.1.3 PCIe Card Removal 94
  - 7.1.4 Environment Failure Protection 94
- 7.2 Module Inspection 94
- 7.3 Environment Failure Protection 95
- 7.4 Module Coatings 95

8 Non-invasive security 97

9 SSP Management 98
9.1 Sensitive Security Parameter 98
9.2 Non-Deterministic Random Bit Generation Specification 118
9.3 Key Import/Export methods 118

10 Self-Tests 121
10.1 Pre-Operational tests 121
10.2 Conditional tests 121
10.3 Periodic Self-Tests 127

11 Life-cycle Assurance 128
11.1 Choosing a secure location for the module 128
11.2 Performing secure initialization of the HSM 128
11.3 Protection of data outside the HSM 129
11.4 Reviewing the Module’s log 130
11.5 Protecting Authentication and Authorization Data 130
11.6 Managing Lost or Stolen iKey 131
11.6.1 User Authentication Tokens 131
11.7 Managing Lost or Stolen Passwords 132
11.7.1 General 132
11.7.2 KCV 132
11.7.3 Remote PED iKeys 132
11.8 Revoking Roles 133
11.9 Key deletion 133
11.10 Decommissioning the HSM 133
11.10.1 Thales Luna PCIe HSM 133
11.10.2 Thales Luna Network HSM 134
11.11 Updating Firmware 135
11.12 Checking Module Status 135
11.13 Maintenance Requirements 135

12 Mitigation of Other Attacks 136

13 Guidance 137
13.1 Identifying the Module Version 137
13.2 Approved Mode of Operation 139
13.3 Using CA_PerformSelftest 140
13.4 Nominal Ranges 141
13.5 Assuming Roles 141
13.6 Additional Guidance 142

ACRONYMS AND ABBREVIATIONS

Term	Definition
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
AU	Audit User
API	Application Programming Interface
CBC	Cipher Block Chaining
CDH	Cofactor Diffie-Hellman
CID	Client IDentity
CITS	Chrysalis ITS
CKG	Cryptographic Key Generation
CFB	Cipher FeedBack
CMAC	Cipher Block Chaining Message Authenticate Code
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CPV1	Cloning Protocol Version 1
CPV3	Cloning Protocol Version 3
CPV4	Cloning Protocol Version 4
CSP	Critical Security Parameter
CTR	CounTeR
CU	Crypto User
CVL	Component Validation List
DAK	Device Authentication Key
DAC	Device Authentication Certificate
DEK	Data Encryption Key

Term	Definition
DH	Diffie-Hellman
DMK	Data MAC Key
DPK	Data Protection Key
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EFP	Environmental Failure Protection
EFT	Environmental Failure Testing
EKA	Ephemeral Key Agreement
ELF	Executable and Linkable Format
EMC	ElectroMagnetic Compatibility
EMI	ElectroMagnetic Interference
ESV	Entropy Source Validation
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
FM	Functionality Module
GCM	Galois Counter Mode
GMAC	Galois Message Authentication Code
GSK	Global Storage Key
HMAC	Keyed-Hash Message Authentication Code
HA	High Availability
HOC	Hardware Origin Certificate
HOK	Hardware Origin Key

Term	Definition
HSE-BBRAM	High-speed erase battery backed RAM
HSM	Hardware Security Module / Host Security Module
ICD	Interface Control Design/Document
IG	Implementation Guidance
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
I/O	Input/Output
IV	Initialization Vector
KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-Based Key Derivation Function
KCV	Key Cloning Vector
KDF	Key Derivation Function
KDM	Key Destruction Method
KEK	Key Encryption Key
KEV	Key Encryption Vector
KTS	Key Transport Scheme
KW	Key Wrap
KWP	Key Wrap with Padding
LCO	Limited Crypto Officer
LED	Light Emitting Diode
LSB	Least Significant Bit
MAC	Message Authentication Code
Masking	A Thales term to describe the encryption of a key for use only within a Thales cryptographic module
Mbps	Megabits per second
MGF	Mask Generation Function

Term	Definition
MIC	Manufacturer’s Integrity Certificate
MIK	Manufacturer’s Integrity Key
MK	Master Key
NIST	National Institute of Science and Technology
N/A	Not Applicable
OFB	Output FeedBack
PAC	PED Authentication Certificate
PAK	PED Authentication Key
PBKDF	Password Based Key Derivation Function
PCIe	Peripheral Component Interconnect Express
PCT	Pair-wise Consistency Test
PEC	Password Encryption Certificate
PED	PIN Entry Device
PEK	Password Encryption Key
PKCS	Public-Key Cryptography Standards
POST	Power-on Self-Test
PSK	Partition Storage Key
PSS	Probabilistic Signature Scheme
PST	Periodic Self-Test
RDK	Role Domain Key
RNG	Random Number Generator
RPV	Remote PED Vector
RSA	Rivest Shamir Adleman
RSADP	RSA Decryption Primitive
RSASVE	RSA Secret-Value Encapsulation
RTC	Real Time Clock

Term	Definition
SALK	Secure Audit Logging Key
SHA	Secure Hash Algorithm
SKA	Static Key Agreement
SMFS	Secure Memory File System
SMK	SKS Master Key
SKS	Scalable Key Storage
SO	Security Officer
SSC	Shared Secret Computation
SSP	Sensitive Security Parameter
STC	Secure Trusted Channel
STM	Secure Transport Mode
Triple-DES	Triple Data Encryption Standard
TUK	Token or Module Unwrapping Key
TVK	Token or Module Variable Key
TWC	Token or Module Wrapping Certificate
USB	Universal Serial Bus
USK	User’s Storage Key
VPD	Vital Product Data
XEX	XOR-encrypt-XOR
XOR	eXclusive OR
XTS	XEX Tweakable block cipher ciphertext Stealing

REFERENCES

[FIPS 140-3]	Federal Information Processing Standards Publication 180-4, Security Requirements for Cryptographic Modules, March 2019.
[FIPS 140-3 IG]	NIST, Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program, May 4, 2021.
[FIPS 180-4]	Federal Information Processing Standards Publication 180-4, Secure Hash Standard (SHS), NIST, August 2015.
[FIPS 186-4]	Federal Information Processing Standards Publication 186-4, Digital Signature Standards (DSS), NIST, July 2013.
[FIPS 197]	Federal Information Processing Standards Publication 197, Specification for the Advanced Encryption Standard (AES), November 26, 2001.
[FIPS 198-1]	Federal Information Processing Standards Publication 198-1, The Keyed-Hash Message Authentication Code (HMAC), July 2008.
[FIPS 202]	Federal Information Processing Standards Publication 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August 2015.
[RFC 5639]	Lochter M, Merkle J, 'Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation', Internet Engineering Task Force, RFC 5639, March 2010.
[RFC 7748]	Hamburg M, Turner S, "Elliptic Curves for Security", Internet Research Task Force, RFC 7748, January 2016.
[SEC 2]	Certicom Research, 'Standards for Efficient Cryptography - SEC2: Recommended Elliptic Curve Domain Parameters', Version 2.0, January 27, 2010.
[SP800-38A]	NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation – Methods and Techniques, December 2001.
[SP800-38B]	NIST Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication, May 2005 (with October 2016 updates).
[SP800-38D]	NIST Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.
[SP800-38E]	NIST Special Publication 800-38E, Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices, January 2010.
[SP800-38F]	NIST Special Publication 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, December 2012.
[SP800-56Ar3]	NIST Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, Revision 3, April 2018.
[SP800-56Br2]	NIST Special Publication 800-56B, Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography, Revision 2, March 2019.

[SP800-56Cr2] NIST Special Publication 800-56C, Recommendation for Key-Derivation Methods in Key-Establishment Schemes, Revision 2, August 2020.

[SP800-67r2] NIST Special Publication 800-67, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Revision 2, November 2017.

[SP800-90Ar1] NIST Special Publication SP800-90A, Recommendation for Random Number Generation Using Deterministic Bit Generators, Revision 1, June 2015.

[SP800-90B] NIST, SP800-90B, “Recommendation for the Entropy Sources Used for Random Bit Generation”, January 2018.

[SP800-108r1] NIST Special Publication 800-108 revision 1, Recommendation for Key Derivation Using Pseudorandom Functions, August 2022.

[SP800-131Ar2] NIST Special Publication 800-131A revision 2, Transitioning the Use of Cryptographic Algorithms and Key Lengths, March 2019.

[SP800-132] NIST Special Publication 800-132, Recommendation for Password-Based Key Derivation: Part 1: Storage Applications, December 2010.

[SP800-133] NIST Special Publication 800-133 revision 2, Recommendation for Cryptographic Key Generation, June 2020.

[SP800-135r1] NIST Special Publication 800-135, Recommendation for Existing Application-Specific Key Derivation Functions, December 2011.

[SP800-140Cr1] NIST Special Publication 800-140C revision 1, CMVP Approved Security Functions: CMVP Validation Authority Updates to ISO/IEC 24759, May 2022.

[SP800-140Dr1] NIST Special Publication 800-140D revision 1, CMVP Approved Sensitive Security Parameter Generation and Establishment Methods: CMVP Validation Authority Updates to ISO/IEC 24759, May 2022.

[SP800-140E] 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.

[SP800-140F] NIST Special Publication 800-140F, CMVP Approved Non-Invasive Attack Mitigation Test Metrics: CMVP Validation Authority Updates to ISO/IEC 24759, March 2020.

[PKCS #1] PKCS #1: RSA Cryptographic Standard, RSA Laboratories, v2.1.

[ANSI X9.42] American National Standard for Financial Services X9.42-2003 (R2013), Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography.

[ANSI X9.62] American National Standard Institute ANSI X9.62, ‘Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA)’, November 16, 2005.

[ANSI X9.63] American National Standard for Financial Services X9.63-2011 (R2017), Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography.

[ISO/IEC 14888-3:2018] ISO/IEC 14888-3:2018, ‘IT Security techniques – Digital Signatures with appendix – Part 3: Discrete logarithm based mechanisms’, 2018-11.

[ISO/IEC 19790:2012] ISO/IEC 19790:2012 (Corrected 2015-12-15, IDT) Information technology – Security techniques – Security requirements for cryptographic modules, 2015-12-15.

[ISO/IEC 24759:2017] ISO/IEC 24759:2017 (Corrected 2017-03, IDT) Information technology – Security techniques – Test requirements for cryptographic modules, 2017-03.

PREFACE

This document deals only with operations and capabilities of the Thales Luna K7 Cryptographic Module in the technical terms of [FIPS 140-3].

General information on Thales HSM alongside other Thales products is available from the following sources:

- the Thales internet site contains information on the full line of available products at https://cpl.thalesgroup.com
- product manuals and technical support literature is available from the Thales Customer Support Portal at https://supportportal.thalesgroup.com/csm
- online manuals for the product can be found at https://www.thalesdocs.com
- technical or sales representatives of Thales can be contacted through one of the channels listed on https://cpl.thalesgroup.com/contact-us

> NOTE: You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

1 General

1.1 Security Level

The Thales Luna K7 Cryptographic Module meets all level 3 security requirements for [FIPS 140-3] as summarized in the table below:

Table 1-1: Security Levels

[ISO/IEC 24759:2017] Section 6 [Number Below]	FIPS 140-3 Section Title	Security Level
1	General	3
2	Cryptographic Module Specification	3
3	Cryptographic Module Interfaces	3
4	Roles, Services, and Authentication	3
5	Software/Firmware Security	3
6	Operational Environment	N/A
7	Physical Security	3
8	Non-Invasive Security	N/A
9	Sensitive Security Parameter Management	3
10	Self-Tests	3
11	Life-Cycle Assurance	3
12	Mitigation of Other Attacks	N/A

2 Cryptographic Module Specification

2.1 Module Overview

The Thales Luna K7 Cryptographic Module is a multi-chip embedded hardware security module in the form of a PCIe card, which typically resides within a custom computing or security appliance. The cryptographic module is contained in its own secure enclosure, which provides physical resistance to tampering.

The cryptographic boundary of the module is defined to encompass all components inside the secure enclosure on the PCIe card.

The module must be explicitly configured to operate in an [FIPS 140-3] approved mode of operation using steps outlined in section 13.2 and where these are performed during the secure initialization of the module.

The module only supports a single approved mode of operation and any configuration changes to settings defining that mode will trigger a zeroization of all partition Sensitive Security Parameter (SSP) and require the full reset and re-initialization of the module.

> **NOTE** Thales Luna K7 Cryptographic Module does not support degraded operation as defined in [ISO/IEC 19790:2012].

The module provides secure key generation and storage for symmetric keys and asymmetric key pairs along with support for a broad range of cryptographic services. Access to key material and cryptographic services for users and user application software is provided through the PKCS #11 programming API, which is implemented over the module's proprietary command interface.

The module may host multiple 'user partitions' which are cryptographically separated and are presented as 'virtual tokens' to user applications. A single 'admin partition' exists, which is dedicated to the HSM Security Officer (HSM SO) and Administrator roles. Each partition must be separately authenticated in order to make it available for use.

2.2 Module Description

The cryptographic module, as defined in [ISO/IEC 19790:2012], is a **hardware module** of embodiment **multi-chip embedded**.

> **NOTE** The Thales Luna K7 Cryptographic Module can be used as follows:
>
> \> as a standalone device called the Thales Luna PCIe HSM; or
>
> \> as an embedded device in the Thales Luna Network HSM.

The cryptographic boundary of the module is shown in Figure 2-1 and Figure 2-2 below with the module embedded in the Thales Luna Network HSM shown in Figure 2-3. The cryptographic boundary is defined as the metal enclosure on the top and bottom sides of the PCIe card as outlined. The fans and heatsinks depicted alongside the removable backup battery are inside the cryptographic boundary (i.e. included in the module versioning on the certificate), but excluded from the testing requirements of FIPS 140-3.

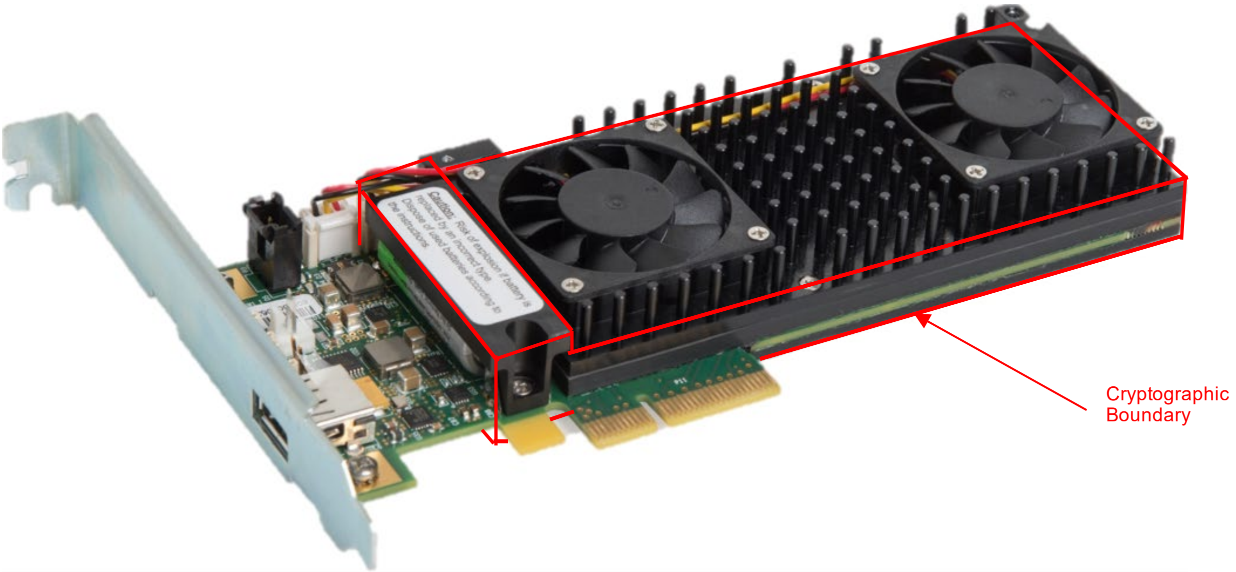

Figure 2-1: Thales Luna K7 Cryptographic Module cryptographic boundary with fans

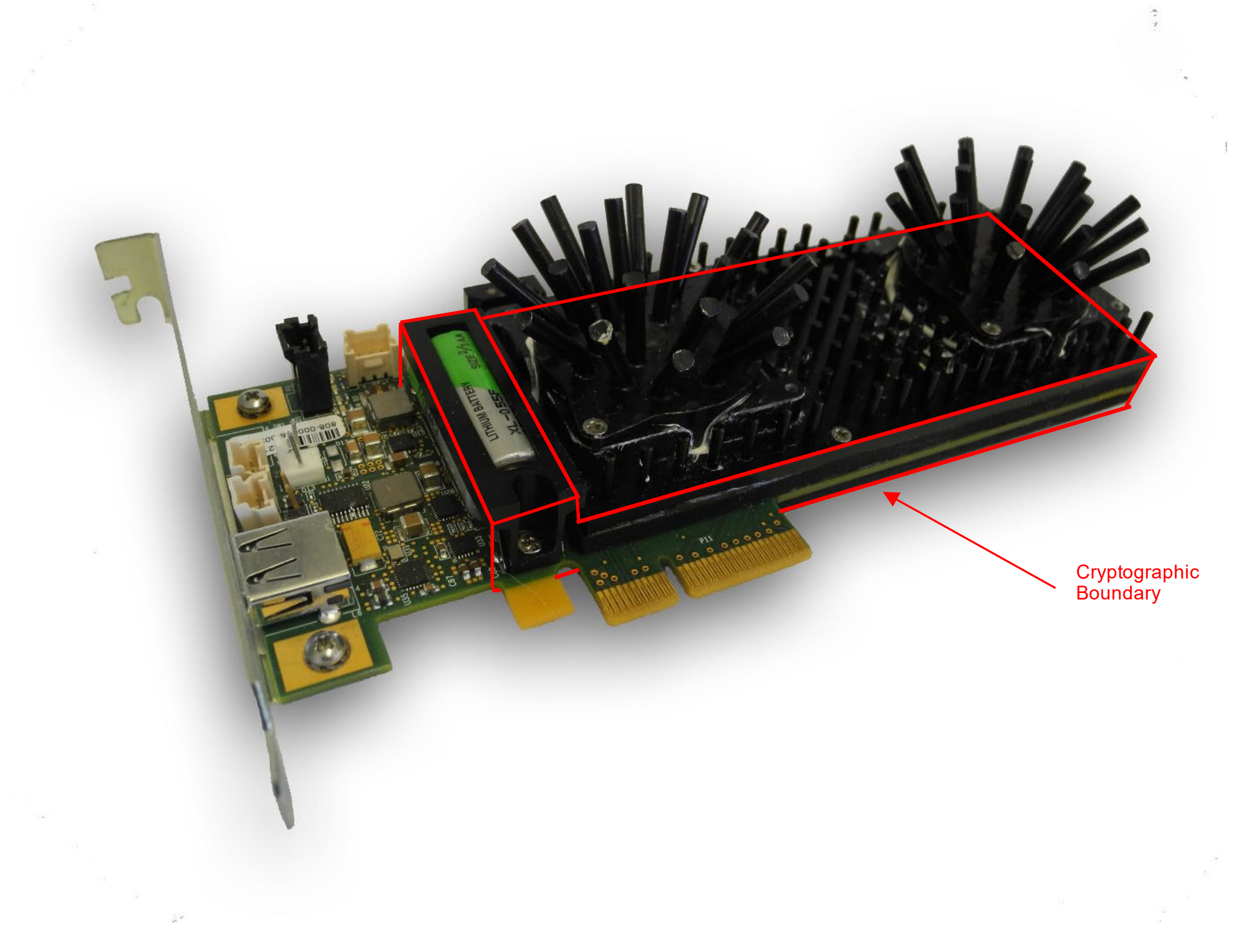

Figure 2-2: Thales Luna K7 Cryptographic Module cryptographic boundary with heatsinks

Figure 2-3: Thales Luna Network HSM

The following figure highlights the logical boundary of the module covered by this certification:

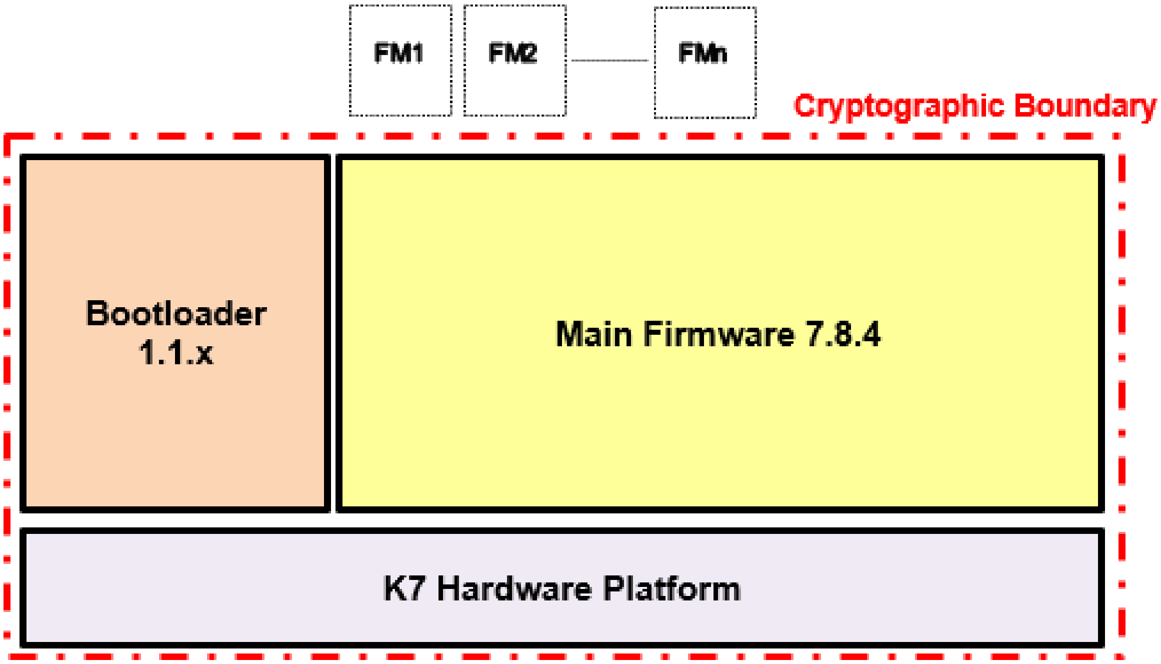

Figure 2-4 – Thales Luna K7 Cryptographic Module, cryptographic boundary.

The boundary includes the bootloader and the main firmware but excludes any Functionality Module (FM) that may be loaded by the HSM SO[1] onto the PCIe card to interface to the main firmware.

The bootloader, main firmware and FM are considered 'firmware' within the scope of definitions in [ISO/IEC 19790:2012].

> **NOTE** As covered above, 'firmware' for the module includes the bootloader, main firmware, and may also contain optional FM's if loaded. To use the module in an approved mode of operation, all firmware, including the bootloader, main firmware and any FM's

[1] The HSM Security Officer is responsible for managing the HSM. As such, the HSM SO is authorized to install and configure the HSM, set and maintain global HSM security policies and load FMs. For more information on the HSM SO, and all other roles, refer to section 4.1.

> loaded onto the module must be validated to [FIPS 140-3] to run on Thales Luna K7 Cryptographic Module.
>
> The scope of this certificate exclusively covers the bootloader and main firmware as approved for use by the module.
>
> Any FM approved to run on the module will have its own FIPS 140-3 certificate to run on top of the module and where it will most likely be of embodiment, **firmware module** if maintaining a FIPS 140-3, Level 3 approval overall.

2.3 Test Configuration

The following tested configuration are covered in this security policy:

Table 2-1: Cryptographic module tested configuration.

Model	Hardware [Part Number and Version]	Firmware Version	Distinguishing Features
Thales Luna PCIe HSM	808-000048-002 or 808-000048-003	Main firmware: 7.8.4 or 7.8.5; with Bootloader: 1.1.1, 1.1.2, 1.1.4, or 1.1.5	Half-height PCIe card with factory installed fans. Listed hardware parts are functionally identical with the difference being a change of supplier for one of the internal components.
Thales Luna K7 Cryptographic Module as used in Thales Luna Network HSM	808-000066-001	Main firmware: 7.8.4 or 7.8.5; with Bootloader: 1.1.1, 1.1.2, 1.1.4, or 1.1.5	Half-height PCIe card with factory installed heatsinks.
Thales Luna K7 Cryptographic Module as used in Thales Luna Network HSM	808-000073-001 or 808-000073-002	Main firmware: 7.8.4 or 7.8.5; with Bootloader: 1.1.1, 1.1.2, 1.1.4, or 1.1.5	Half-height PCIe card with factory installed heatsinks. This model is physically identical to 808-000066-001. Listed hardware parts are functionally identical with the difference being a change of supplier for one of the internal components.

This document covers both the PED and password authentication configurations of the Thales Luna K7 Cryptographic Module.

NOTE The security features described in this document apply to the Thales Luna K7 Cryptographic Module only and do not include any feature that may be enforced by the host appliance, client or Thales Luna PED.

NOTE As the module is a hardware module of embodiment **multi-chip embedded** – this security policy is not required to list an operating system.

2.4 Approved Algorithms

The following cryptographic library and associated CAVP certificates are used by the cryptographic module:

> **SafeNet Bootloader Cryptographic Library** (Cert #C1701 and #A3164);

> **SafeNet Accelerated Cryptographic Library** (Certs #C1707 and #A480);

> **SafeNet Accelerated Cryptographic Library – Alternate** (Cert #C1717 and #A481);

> **SafeNet Cryptographic Library** (Cert #C1718 and #A478); and

> **SafeNet Cryptographic Library – Alternate** (Cert #C1719 and #A479).

NOTE The following certificates referenced above contain redundant listings not used by the cryptographic module:

> Cert #C1701 includes a listing for FIPS 186-2, RSA, SigVer. This is redundant listing and where the same implementation has been retested to [FIPS 186-4], RSA, SigVer covered under Cert #A3164.

> Cert #C1707 includes listing for KAS-ECC, KAS-FFC, CVL (RSADP), GMAC and TDES-CMAC as a redundant MAC option for KDF [SP800-108r1];

> Cert #C1717 includes a listing for KAS-ECC and CVL (RSADP);

> Cert #C1718 includes KAS-ECC, KAS-FFC, GMAC, CVL (RSADP) and AES in GCM mode with external IV; and

> Cert #C1719 includes CVL (RSADP).

Implementations for these algorithms are present in the software libraries or hardware integrated circuits used by the module and have completed CAVP testing but where **this code is not executable for the certified configuration of the module**.

The following entropy source and associated ESV certificated are used by the cryptographic module:

> **Thales K7 Hardware Platform TRNG** (ESV Cert #98).

The approved algorithms implemented by the module alongside their mapping to the certificates above alongside algorithms use by service are listed in the Table 2-2 below.

Listings in the ‘Use / Function’ column map to services listed in Table 4-2 and Table 4-4:

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Symmetric Encryption/Decryption				
#C1707	**Algorithm:** AES. **Standards:** [FIPS 197], [SP800-38A], [SP800-38D], [SP800-38E] and [SP800-38F].	**Mode:** CBC, CFB128, CFB8, CTR, ECB, GCM[2], KW, KWP[3], OFB, XTS[4].	**Key size:** 128 and 256-bit - all modes. 192-bits – CBC, CFB128, CFB8, CTR, ECB, GCM, KW, KWP AND OFB only.	Initialize the HSM, Create a user partition, Export/import audit log secret key, Request partition STC identity, Initiate STC tunnel, Send commands to partition with STC tunnel initiated, Clone SMK between partitions, Clone partition objects between partitions, Rollover SMK for a given partition, Request HSM self-test, Initialize role, Configure partition for high-available recovery / login, Login as role, Initialize Remote PED Vector (RPV), Send or receive data over PED tunnel (local PED), Send or receive data over PED tunnel (remote PED), Generate local symmetric or asymmetric key-pair, Generate domain parameters, Derive key from existing partition secret or private key object, Import secret or private key using key wrapping, Export secret or private key using key wrapping, Insert key from external storage using SKS, Extract key to external storage using SKS, Perform encrypt operation on user supplied data object, Perform decrypt operation on user supplied data object, Generate signature or MAC over user supplied data, Activate SMFS, Store/retrieve data from SMFS.
#C1718	**Algorithm:** AES. **Standards:** [FIPS 197], [SP800-38D] and [SP800-38F].	**Mode:** KW, KWP.	**Key size:** 128, 192 and 256-bit.	*<As per #C1707 'AES' services above using KW and KWP, but used when data being encrypted or decrypted is greater than 2KB in size.>*
#C1707	**Algorithm:** Triple-DES. **Standards:** [SP800-67r2] and [SP800-38A].	**Mode:** CBC, CFB64, CFB8, CTR, ECB, OFB (decrypt only).	**Key size:** 168-bits (3-key).	Request HSM self-test, Import secret or private key using key wrapping, Perform decrypt operation on user supplied data object, Validate signature or MAC over user supplied data.

[2] The module generates IVs internally using the approved DRBG where all IV used are 128-bits in length per [SP800-38D].

[3] KW and KWP use this implementation configuration when objects are less than 2KB in size. Otherwise, the #C1718 implementation is used.

[4] XTS-AES is only supported for consumption by users with the 'Perform encrypt operation on user supplied data object' service. When used, output from this service should be used exclusively for data storage. Keys generated using PBKDF are exclusively used for functions internal to the module and where these are used for storage applications.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Hashing				
#C1707	**Algorithm:** SHA. **Standards:** [FIPS 186-4] and [FIPS 202].	**Methods:** SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE-128, SHAKE-256 (Byte Only).	N/A.	Update firmware, Protect object integrity, Load configuration update file, Request partition STC identity, Initiate STC tunnel, Clone SMK between partitions, Clone partition objects between partitions, Enable/disable STM, Request HSM self-test, Initialize role, Change authentication data, Configure partition for high-available recovery / login, Login as role, Initialize Remote PED Vector (RPV), Generate local symmetric or asymmetric key-pair, Generate domain parameters, Derive key from existing partition secret or private key object, Import secret or private key using key wrapping, Export secret or private key using key wrapping, Insert key from external storage using SKS, Re-seed partition DRBG, Extract entropy from partition DRBG, Perform digest operation on user supplied data, Perform encrypt operation on user supplied data object, Generate signature or MAC over user supplied data, Download FM, Activate SMFS, Store/retrieve data from SMFS, Setup Local PED Session, Setup Remote PED Session, Send or receive data over PED tunnel (remote PED).
#C1718	**Algorithm:** SHA. **Standards:** [FIPS 180-4].	**Methods:** SHA2-256, SHA2-512 (Byte Only).	N/A.	Generate secure log record, Submit external messages for entry into secure audit log, Validate the audit log, Request HSM self-test.
#C1701	**Algorithm:** SHA. **Standards:** [FIPS 180-4].	**Methods:** SHA-1, SHA2-384 (Byte Only).	N/A.	Request authentication and execution of main firmware.
Message Authentication Code				
#C1707	**Algorithm:** HMAC. **Standard:** [FIPS 198-1].	**Methods:** HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512.	**Mac size:** 10-to-64 bytes (dependent on hash). **Key size:** key size < block size, key size = block size, key size > block size.	Request HSM self-test, Send or receive data over PED tunnel (remote PED), Generate signature or MAC over user supplied data, Validate signature or MAC over user supplied data, Store/retrieve data from SMFS, Initialize the HSM, Initialize role, Change authentication data, Login as role
#C1718	**Algorithm:** HMAC. **Standard:** [FIPS 198-1].	**Methods:** HMAC-SHA2-256.	**Mac size:** 16, 24, 32 bytes. **Key size:** key size < block size, key size = block size, key size > block size.	*<as per SHA from Cert #C1718 covered above where this certificate covers the standalone algorithm implementation exclusively used to support secure logging.>*

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C1707	**Algorithm:** Triple-DES. **Standard:** [SP800-67r2] and [SP800-38B].	**Methods:** CMAC (MAC validate only).	**Key size:** 168-bits (3-key).	Validate signature or MAC over user supplied data, Request HSM self-test.
#C1707	**Algorithm:** AES. **Standard:** [FIPS 197] and [SP800-38B].	**Methods:** CMAC.	**Key size:** 128, 192, 256-bits.	Generate signature or MAC over user supplied data, Validate signature or MAC over user supplied data, Request HSM self-test.
Asymmetric				
#C1707, #C1717.	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Method:** Key Generation, Signature Generation, Signature Verification. **Signature Type:** ANSI X9.31, PKCS #1-v1.5 1.5, PKCS-PSS. **Hash options:** Signature Generation (PKCS #1-v1.5 and PKCS-PSS): SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Generation (ANSI X9.31): SHA2-224, SHA2-256, SHA2-384, SHA2-512. Signature Verification (PKCS #1-v1.5 and PKCS-PSS): SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification (ANSI X9.31): SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512. **Vendor affirmed using [FIPS 140-3 IG], C.C, The Use and the Testing Requirements for the Family of Functions defined in FIPS 202, when using SHA-3.**	**Modulus length:** 2048, 3072 – Key Generation, Signature Generation and Signature Verification. 1024 – Signature Verification only.	Clone SMK between partitions, Clone partition objects between partitions, Generate local symmetric or asymmetric key-pair, Request HSM self-test, Generate signature or MAC over user supplied data, Validate signature or MAC over user supplied data, Download FM. Both algorithm implementations used when **HSM Policy (43) Enable low-level math acceleration** is `enabled`. Of the two implementations, Algorithm implementation covered by #C1707 used when **Partition Policy (16) Operate Without RSA blinding** is `enabled` and #C1717 used when **Partition Policy (16) Operate Without RSA blinding** is `disabled`.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C1718, #C1719	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Method:** Key Generation. **Key Generation Method:** [FIPS 186-4] B.3.3 and B.3.6.	**Modulus length:** 2048 and 3072-bit.	Used for Generate local symmetric or asymmetric key-pair as already mapped from the RSA listing above but where this implementation is used when **HSM Policy (43) Enable low level math acceleration** is `disabled`. Of the two implementations, Algorithm implementation covered by Cert #C1718 used when **Partition Policy (16) Operate Without RSA blinding** is `enabled` and Cert #C1719 used when **Partition Policy (16) Operate Without RSA blinding** is `disabled`.
#A480, #A481.	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Methods:** Signature Generation, Signature Verification. **Signature Type:** ANSI X9.31, PKCS #1-v1.5, PKCS-PSS. Signature Generation (PKCS #1-v1.5 and PKCS-PSS): SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Generation (ANSI X9.31): SHA2-224, SHA2-256, SHA2-384, SHA2-512. Signature Verification (PKCS #1-v1.5 and PKCS-PSS): SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification (ANSI X9.31): SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512. **Vendor affirmed using [FIPS 140-3 IG], C.C, The Use and the Testing Requirements for the Family of Functions defined in FIPS 202, when using SHA-3.**	**Modulus length:** 4096-bit. Vendor Note: Key sizes above 4096 and up to modulus length 8192-bit are supported for signature generation and verification by the module as permitted by [SP800-131Ar2] but were not supported for test by the NIST CAVP program above modulus 4096-bits at the time of module submission. Modulus above 4096-bits may be used in the approved mode but are untested for RSA as part of the independent CAVP assurance activities performed.	Load configuration update file, Clone SMK between partitions, Clone partition objects between partitions, , Configure partition for high-available recovery / login, Generate local symmetric or asymmetric key-pair, Request HSM self-test, Generate signature or MAC over user supplied data, Validate signature or MAC over user supplied data, Download FM. Both algorithm implementations used when **HSM Policy (43) Enable low-level math acceleration** is `enabled`. Of the two implementations, Algorithm implementation covered by Cert #A480 used when **Partition Policy (16) Operate Without RSA blinding** is `enabled` and Cert #A481 used when **Partition Policy (16) Operate Without RSA blinding** is `disabled`.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A478, #A479, #A480, #A481.	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Method:** Key Generation. **Key Generation Methods:** B.3.3 and B.3.6.	**Modulus length:** 4096-bit. Vendor Note: Key sizes above 4096 and up to modulus length 8192-bit are supported for signature generation and verification by the module as permitted by [SP800-131Ar2] but were not supported for test by the NIST CAVP program above modulus 4096-bits at the time of module submission. Modulus above 4096-bits may be used in the approved mode but are untested for RSA as part of the independent CAVP assurance activities performed.	Generate local symmetric or asymmetric key-pair. Implementation covered by Cert #A480 used when **HSM Policy (43) Enable low-level math acceleration** is `enabled` and **Partition Policy (16) Operate Without RSA blinding** is `enabled`. Implementation covered by Cert #A481 used when **HSM Policy (43) Enable low-level math acceleration** is `enabled` and **Partition Policy (16) Operate Without RSA blinding** is `disabled`. Implementation covered by Cert #A478 used when **HSM Policy (43) Enable low-level math acceleration** is `disabled` and **Partition Policy (16) Operate Without RSA blinding** is `enabled`. Implementation covered by Cert #A479 used when **HSM Policy (43) Enable low-level math acceleration** is `disabled` and **Partition Policy (16) Operate Without RSA blinding** is `disabled`.
#A3164	**Algorithm:** RSA. **Standard:** [FIPS 186-4]**.**	**Method:** Signature Verification. **Signature Type:** PKCS #1-v1.5. **Hash options:** SHA-1, SHA2-384.	**Modulus length:** 4096-bit.	Request authentication and execution of main firmware.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C1707	**Algorithm:** DSA. **Standard:** [FIPS 186-4].	**Methods:** Parameter Generation, Key Generation, Signature Generation, Signature Verification. Hash options: Parameter Generation: SHA2-224, SHA2-256. Signature Generation: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification: SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. **Vendor affirmed using [FIPS 140-3 IG], C.C, The Use and the Testing Requirements for the Family of Functions defined in FIPS 202, when using SHA-3.**	**Modulus length:** 2048 and 3072 – Parameter Generation, Key Generation, Signature Generation. 1024, 2048 and 3072 – Signature Verification.	Generate local symmetric or asymmetric key-pair, Generate domain parameters, Generate signature or MAC over user supplied data, Validate signature or MAC over user supplied data.
#C1718	**Algorithm:** DSA. **Standard:** [FIPS 186-4].	**Methods:** Parameter Generation, Key Generation. **Hash Options:** Parameter Generation: SHA2-224, SHA2-256.	**Modulus length:** 2048 and 3072 – Parameter Generation, Key Generation.	Generate local symmetric or asymmetric key-pair, Generate domain parameters when **HSM Policy (43) Enable low-level math acceleration** is `disabled`.
#C1707	**Algorithm:** ECDSA. **Standard:** [FIPS 186-4].	**Methods:** Key Generation, Signature Generation, Signature Verification. **Hash options:** Signature Generation: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification: SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521. Signature Verification additional **Curves**: P-192, K-163, and B-163 Non-NIST (as per [FIPS 140-3] IG C.A): see Table 2-3 below.	Request partition STC identity, Initiate STC tunnel, Generate local symmetric or asymmetric key-pair, Generate signature or MAC over user supplied data, Validate signature or MAC over user supplied data, Initialize Remote PED Vector (RPV). Note: Initiate STC tunnel , Request partition STC identity and Initialize Remote PED Vector (RPV) exclusively use P-521. Algorithm implementation used when **HSM Policy (43) Enable low-level math acceleration** is `enabled.`

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C1718	**Algorithm:** ECDSA. **Standard:** [FIPS 186-4].	**Methods:** Key Generation, Signature Generation, Signature Verification. **Hash options:** Signature Generation: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification: SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571. Signature Verification additional **Curves**: K-163 and B-163 Non-NIST (as per [FIPS 140-3] IG C.A): see Table 2-3 below for curves over a binary field – GF(2^m) exclusively.	As per services mapped to ECDSA for #C1707 above but where this implementation is only exercised for the listed curves and when **HSM Policy (43) Enable low-level math acceleration** is `disabled`.
Key Agreement Scheme				
#A480	**Algorithm:** KAS (KAS-ECC-SSC (Cert #A480) and KDA (Cert #A480)). **Standard:** [SP800-56Ar3] and [SP800-56Cr2].	**Methods:** ephemeralUnified and onePassDH with OneStep KDF from [SP800-56Cr2][5] with Auxiliary function: SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384 or SHA3-512.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521. **Caveat:** key establishment methodology provides between 112 and 256 bits of encryption strength.	Request HSM self-test, Derive key from existing partition secret or private key object. Algorithm implementation used when **HSM Policy (43) Enable low-level math acceleration** is `enabled.`
#A480	**Algorithm:** KAS (KAS-ECC-SSC (Cert #A480) and CVL (Cert #A480)). **Standard:** [SP800-56Ar3] and [SP800-135r1].	**Methods:** ephemeralUnified and onePassDH with X9.63 KDF [SP800-135r1] using SHA2-224, SHA2-256, SHA2-384 or SHA2-512.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521. **Caveat:** key establishment methodology provides between 112 and 256 bits of encryption strength.	Request HSM self-test, Derive key from existing partition secret or private key object. Algorithm implementation used when **HSM Policy (43) Enable low-level math acceleration** is `enabled.`
#A480, #A478	**Algorithm:** KAS (KAS-ECC-SSC (#A478) and KDA (#A480)). **Standard:** [SP800-56Ar3] and [SP800-56Cr2].	**Methods:** ephemeralUnified and onePassDH with OneStep KDF from [SP800-56Cr2][6] with Auxiliary function: SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384 or SHA3-512.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571. **Caveat:** key establishment methodology provides between 112 and 256 bits of encryption strength.	As per services mapped to KAS-ECC-SSC for Cert #A480 above but where this implementation is only exercised for the listed curves and when **HSM Policy (43) Enable low-level math acceleration** is `disabled`.

[5] available for use with the C_DeriveKey ICD command.

[6] available for use with the C_DeriveKey ICD command.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A480, #A478	**Algorithm:** KAS (KAS-ECC-SSC (#A478) and CVL (#A480)). **Standard:** [SP800-56Ar3] and [SP800-135r1].	**Methods:** ephemeralUnified and onePassDH with X9.63 KDF [SP800-135r1] using SHA2-224, SHA2-256, SHA2-384 or SHA2-512.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571. **Caveat:** key establishment methodology provides between 112 and 256 bits of encryption strength.	As per services mapped to KAS-ECC-SSC for Cert #A480 above but where this implementation is only exercised for the listed curves and when **HSM Policy (43) Enable low-level math acceleration** is `disabled`.
#A480	**Algorithm:** KAS (KAS-FFC-SSC (Cert #A480) and KDA (Cert #A480)). **Standard:** [SP800-56Ar3] and [SP800-56Cr2].	**Methods:** dhHybrid1, dhEphem, dhHybridOneFlow and dhOneFlow with OneStep KDF from [SP800-56Cr2] with Auxiliary function: SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384 or SHA3-512.	**Modulus length:** 2048, 3072 and 4096. **Caveat:** key establishment methodology provides between 112 and 150-bits of encryption strength.	Request HSM self-test, Derive key from existing partition secret or private key object. Implementation of KAS-FFC-SSC covered by Cert #A480 is used when **HSM Policy (43) Enable low-level math acceleration** is `enabled`.
#A480	**Algorithm:** KAS (KAS-FFC-SSC (Cert #A480) and CVL (Cert #A480)). **Standard:** [SP800-56Ar3] and [SP800-135r1].	**Methods:** dhHybrid1, dhEphem, dhHybridOneFlow and dhOneFlow with X9.42 KDF from [SP800-135r1] using SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384 or SHA3-512.	**Modulus length:** 2048, 3072 and 4096. **Caveat:** key establishment methodology provides between 112 and 150-bits of encryption strength.	Request HSM self-test, Derive key from existing partition secret or private key object. Implementation of KAS-FFC-SSC covered by Cert #A480 is used when **HSM Policy (43) Enable low-level math acceleration** is `enabled`.
#A480, #A478	**Algorithm:** KAS (KAS-FFC-SSC (Cert. #A478) and KDA (Cert #A480)). **Standard:** [SP800-56Ar3] and [SP800-56Cr2].	**Methods:** dhHybrid1, dhEphem, dhHybridOneFlow and dhOneFlow with OneStep KDF from [SP800-56Cr2] with Auxiliary function: SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384 or SHA3-512.	**Modulus length:** 2048, 3072 and 4096. **Caveat:** key establishment methodology provides between 112 and 150-bits of encryption strength.	Request HSM self-test, Derive key from existing partition secret or private key object. Implementation of KAS-FFC-SSC covered by Cert #A478 is used when **HSM Policy (43) Enable low-level math acceleration** is `disabled`.
#A480, #A478	**Algorithm:** KAS (KAS-FFC-SSC (Cert. #A478) and CVL (#A480)). **Standard:** [SP800-56Ar3] and [SP800-135r1].	**Methods:** dhHybrid1, dhEphem, dhHybridOneFlow and dhOneFlow with X9.42 KDF from [SP800-135r1] using SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384 or SHA3-512.	**Modulus length:** 2048, 3072 and 4096. **Caveat:** key establishment methodology provides between 112 and 150-bits of encryption strength.	Request HSM self-test, Derive key from existing partition secret or private key object. Implementation of KAS-FFC-SSC covered by Cert #A478 is used when **HSM Policy (43) Enable low-level math acceleration** is `disabled`.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A480	**Algorithm:** KAS-ECC. **Standards:** [SP800-56Ar3] and [SP800-56Cr2].	**Methods:** onePassDH with full key validation and key pair generation **KDF:** OneStep Key Derivation using SHA2-512. **Key Confirmation:** HMAC-SHA2-512 with 256-bit key and 512-bit MAC.	**Curve:** P-521.	Setup Local PED Session.
#A480	**Algorithm:** KAS-ECC. **Standards:** [SP800-56Ar3] and [SP800-56Cr2].	**Methods:** fullUnified with full key validation and key-pair generation **KDF:** OneStep using SHA2-512. **Key Confirmation:** HMAC-SHA2-512 with 256-bit key and 512-bit MAC.	**Curve:** P-521.	Setup Remote PED Session.
#A480	**Algorithm:** KAS-ECC. **Standards:** [SP800-56Ar3] and [SP800-56Cr2].	**Methods:** fullUnified with full key validation and key pair generation **KDF:** OneStep using SHA2-512. **Key Confirmation:** HMAC with 512-bit key and 512-bit MAC.	**Curve:** P-521.	Initiate STC tunnel.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A478, #A479, #A480, #A481	**Algorithm:** KAS-RSA. **Standards:** [SP800-56Br2] and [SP800-56Cr2]	**Method:** KAS1-basic. **Key generation method:** rsakpg1-crt, rsakpg2-crt **KDF method**: One-Step Key Derivation from [SP800-56Cr2] using SHA2-512.	**Modulus length:** 4096-bit.	Clone SMK between partitions, Configure partition for high-available recovery / login. Implementation covered by Cert #A480 used when **HSM Policy (43) Enable low-level math acceleration** is `enabled` and **Partition Policy (16) Operate Without RSA blinding** is `enabled`. Implementation covered by Cert #A481 used when **HSM Policy (43) Enable low-level math acceleration** is `enabled` and **Partition Policy (16) Operate Without RSA blinding** is `disabled`. Implementation covered by Cert #A478 used when **HSM Policy (43) Enable low-level math acceleration** is `disabled` and **Partition Policy (16) Operate Without RSA blinding** is `enabled`. Implementation covered by Cert #A479 used when **HSM Policy (43) Enable low-level math acceleration** is `disabled` and **Partition Policy (16) Operate Without RSA blinding** is `disabled`.
#A480	**Algorithm:** KAS-ECC-SSC. **Standards:** [SP800-56Ar3].	**Methods:** ephemeralUnified, onePassDH.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521.	Request HSM self-test, Derive key from existing partition secret or private key object. Algorithm implementation used when **HSM Policy (43) Enable low-level math acceleration** is `enabled.`
#A478	**Algorithm:** KAS-ECC-SSC. **Standards:** [SP800-56Ar3].	**Methods:** ephemeralUnified, onePassDH.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571.	As per services mapped to KAS-ECC-SSC for Cert #A480 above but where this implementation is only exercised for the listed curves and when **HSM Policy (43) Enable low-level math acceleration** is `disabled`.
#A478, #A480	**Algorithm:** KAS-FFC-SSC. **Standards:** [SP800-56Ar3].	**Methods:** dhHybrid1, dhEphem, dhHybridOneFlow, dhOneFlow.	**Modulus length:** 2048, 3072 and 4096-bit.	Request HSM self-test, Derive key from existing partition secret or private key object. Implementation covered by Cert #A480 is used when **HSM Policy (43) Enable low-level math acceleration** is `enabled`. Implementation covered by Cert #A478 is used when **HSM Policy (43) Enable low-level math acceleration** is `disabled`.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Key Transport				
#C1707	**Algorithm:** KTS (AES Cert #C1707). **Standards:** [FIPS 197], [SP800-38D] and [SP800-38F].	**Modes:** GCM, KW and KWP.	**Key size:** 128, 192, and 256-bit. **Caveat:** key establishment methodology provides between 128 and 256 bit of encryption strength.	Export/import audit log secret key, Configure partition for high-available recovery / login, Export secret or private key using key wrapping, Clone SMK between partitions, Clone partition objects between partitions. AES KW and AES KWP implementation covered by this KTS is used when data sizes being encrypted or decrypted are less than 2KB in size. When cloning objects using CPV3, KWP is used. When cloning objects with CPV4, GCM is one of two supported key transport options (CTR with HMAC-SHA2-512 is the other, as covered below).
#C1718	**Algorithm:** KTS (AES Cert #C1718). **Standards:** [FIPS 197] and [SP800-38F].	**Modes:** KW and KWP.	**Key size:** 128, 192, and 256-bit. **Caveat:** key establishment methodology provides between 128 and 256 bit of encryption strength.	Export/import audit log secret key, Configure partition for high-available recovery / login, Export secret or private key using key wrapping, Clone SMK between partitions, Clone partition objects between partitions. AES KW and AES KWP implementation covered by this KTS are exclusively used when data sizes being encrypted or decrypted are 2KB or greater in size. When cloning objects using CPV3, KWP is used.
#C1707.	**Algorithm:** KTS (AES Cert #C1707, HMAC Cert #C1707). **Standards:** [FIPS 197], [FIPS 198-1] and [SP800-38F].	**Modes:** CTR. **MAC:** HMAC-SHA2-512 or HMAC-SHA3-512.	**Key Size:** 256-bits (separate encryption and MAC keys, each of 256-bit).	Clone SMK between partitions, Clone partition objects between partitions – both when exclusively using the Cloning Protocol Version 4. CTR with HMAC-SHA2-512 is one key transport option available with CPV4 with the other option being GCM covered above.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A478, #A479, #A480, #A481	**Algorithm:** KTS-RSA. **Standards:** [SP800-56Br2] and [SP800-56Cr2].	**Method:** KTS-OAEP-basic. **Key generation method:** rsakpg1-crt and rsakpg2-crt. **Hash:** SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. **Mask Generation Function:** SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	**Modulus length:** 2048, 3072, 4096, 6144, and 8192. **Caveat:** key establishment methodology provides between 112 and 201 bits of encryption strength.	Request HSM self-test, Import secret or private key using key wrapping. Implementation covered by Cert #A480 used when **HSM Policy (43) Enable low-level math acceleration** is `enabled` and **Partition Policy (16) Operate Without RSA blinding** is `enabled`. Implementation covered by Cert #A481 used when **HSM Policy (43) Enable low-level math acceleration** is `enabled` and **Partition Policy (16) Operate Without RSA blinding** is `disabled`. Implementation covered by Cert #A478 used when **HSM Policy (43) Enable low-level math acceleration** is `disabled` and **Partition Policy (16) Operate Without RSA blinding** is `enabled`. Implementation covered by Cert #A479 used when **HSM Policy (43) Enable low-level math acceleration** is `disabled` and **Partition Policy (16) Operate Without RSA blinding** is `disabled`.
Key Derivation Function,				
#C1707	**Algorithm:** Key-Based Key Derivation Function (KBKDF). **Standards:** [SP800-108r1].	**Mode:** Counter. **MAC Mode:** CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512.	**Supported Lengths:** 1024, 1032, 2048, and 2056. **Fixed Data Order:** Before Fixed Data. **Counter Length:** 32.	Initialize the HSM, Initialize role, Derive key from existing partition secret or private key object, Change authentication data, Login as role, Request HSM self-test.
#A480	**Algorithm:** KDA[7]. **Standards:** [SP800-56Cr2].	**Method:** One-Step Key Derivation. **Hash:** SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	**Shared secret length:** 224-8192, increment 1 byte. **Derived Key length:** 128 – 4096-bits, increment 1 byte.	Request HSM self-test, Derive key from existing partition secret or private key object, Clone SMK between partitions, Clone partition objects between partitions.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A480	**Algorithm:** X9.42 Key Derivation Algorithm (CVL). **Standards:** [SP800-133], [SP800-135r1] and [ANSI X9.42].	**Methods:** SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	**Shared secret length:** 64-4096-bits, increment 1 byte. **Derived Key Length:** 384-bits.	Request HSM self-test, Derive key from existing partition secret or private key object.
#A480	**Algorithm:** X9.63 Key Derivation Function (CVL). **Standards:** [ANSI X9.63].	**Methods:** SHA2-224, SHA2-256, SHA2-384, SHA2-512.	**Field Size:** 224, 256, 384, 521. **Shared Secret Length**: 128-4096 Increment 8-bits.	Perform encrypt operation on user supplied data object
#A480	**Algorithm:** PBKDF[8]. **Standards:** [SP800-132].	**Methods:** HMAC-SHA2-512 (as covered under Cert #C1707).	**Derived Key Length:** 256-bit. **Password Length:** 128-bit. **Salt Length:** 256-bits.	Initialize the HSM, Initialize role, Change authentication data, Login as role, Request HSM self-test.
Random Number Generation				
ESV (Cert #98)	**Algorithm:** Physical. **Standards:** [SP800-90B], [FIPS 180-4].	**Methods:** Live noise source with SHA2-512 vetted conditioning function.	**Security Strength:** Full Entropy.	Initialize the HSM, Create a user partition, Request partition STC identity, Initiate STC tunnel, Clone SMK between partitions, Clone partition objects between partitions, Rollover SMK for a given partition, Enable/disable STM, Request HSM self-test, Initialize role, Configure partition for high-available recovery / login, Initialize Remote PED Vector (RPV), Setup Local PED Session, Setup Remote PED Session, Generate local symmetric or asymmetric key-pair, Generate domain parameters, Derive key from existing partition secret or private key object, Export secret or private key using key wrapping, Re-seed partition DRBG, Extract entropy from partition DRBG, Perform encrypt operation on user supplied data object, Generate signature or MAC over user supplied data, Activate SMFS
#C1707	**Algorithm:** CTR_DRBG. **Standard:** [SP800-90Ar1].	**Mode:** AES-256.	**Security strength:** 256-bit.	<as per ESV (Cert #98) on row above>

[8] Used internal to the cryptographic module to derive the storage encryption key used to encrypt the checkword used during password based authentication. The derived key is separately used to encrypt for storage the USK which is independently also encrypted under the module generated KEK. The module uses method 1a from [SP800-132] where the derived Master Key (MK) is used directly as the Data Protection Key (DPK).

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Key Generation				
Vendor Affirmed	**Algorithm:** CKG. **Standard:** [SP800-133].	**Method:** symmetric keys and seed for asymmetric key generation are created based on the direct output of the module DRBG (Cert #C1707).	**Security strength:** 256-bit.	Initialize the HSM, Create a user partition, Request partition STC identity, Initiate STC tunnel, Clone SMK between partitions, Clone partition objects between partitions, Rollover SMK for a given partition, Initialize role, Change authentication data, Initialize Remote PED Vector (RPV), Setup Remote PED Session, Generate local symmetric or asymmetric key-pair, Generate domain parameters, Activate SMFS.

Table 2-3: Supported non-NIST elliptic curve as per [FIPS 140-3] IG C.A

Curve Name	Curve Field Type	Definition	Security Strength	Permitted Operations		
				Sign	Verify	Derive
sect571r2	Binary field - GF(2^m)	[SEC 2].	285-bit	x	x	x
sect571k2	Binary field - GF(2^m)	[SEC 2].	285-bit	x	x	x
Brainpool P512r1	Prime field – GF(p)	[RFC 5639].	256-bit	x	x	x
Brainpool P512t1	Prime field – GF(p)	[RFC 5639].	256-bit	x	x	x
X9.62 c2pnb431r1	Binary field - GF(2^m)	[ANSI X9.62].	215-bit	x	x	x
sect409r1	Binary field - GF(2^m)	[SEC 2].	204-bit	x	x	x
sect409k1	Binary field - GF(2^m)	[SEC 2].	204-bit	x	x	x
Brainpool P384r1	Prime field – GF(p)	[RFC 5639].	192-bit	x	x	x
Brainpool P384t1	Prime field – GF(p)	[RFC 5639].	192-bit	x	x	x
X9.62 c2pnb368w1	Binary field - GF(2^m)	[ANSI X9.62].	184-bit	x	x	x
X9.62 c2pnb359v1	Binary field - GF(2^m)	[ANSI X9.62].	179-bit	x	x	x
Brainpool P320r1	Prime field – GF(p)	[RFC 5639].	160-bit	x	x	x
Brainpool P320t1	Prime field – GF(p)	[RFC 5639].	160-bit	x	x	x
X9.62 c2pnb304w1	Binary field - GF(2^m)	[ANSI X9.62].	152-bit	x	x	x
sect283r1	Binary field - GF(2^m)	[SEC 2].	141-bit	x	x	x
sect283k1	Binary field - GF(2^m)	[SEC 2].	141-bit	x	x	x
X9.62 c2pnb272w1	Binary field - GF(2^m)	[ANSI X9.62].	136-bit	x	x	x
sm2p256v1	Prime field – GF(p)	[ISO/IEC 14888-3:2018]	128-bit	x	x	x
secp256k1	Prime field – GF(p)	[SEC 2].	128-bit	x	x	x
Brainpool P256r1	Prime field – GF(p)	[RFC 5639].	128-bit	x	x	x
Brainpool P256r1	Prime field – GF(p)	[RFC 5639].	128-bit	x	x	x
Curve25519	Prime field – GF(p)	[RFC 7748]	128-bit	x	x	x
X9.62 prime239v3	Prime field – GF(p)	[ANSI X9.62].	119-bit	x	x	x
X9.62 prime239v2	Prime field – GF(p)	[ANSI X9.62].	119-bit	x	x	x
X9.62 prime239v1	Prime field – GF(p)	[ANSI X9.62].	119-bit	x	x	x
X9.62 c2pnb239v1	Binary field - GF(2^m)	[ANSI X9.62].	119-bit	x	x	x
X9.62 c2pnb239v2	Binary field - GF(2^m)	[ANSI X9.62].	119-bit	x	x	x

Curve Name	Curve Field Type	Definition	Security Strength	Permitted Operations		
				Sign	Verify	Derive
X9.62 c2pnb239v3	Binary field - GF(2^m)	[ANSI X9.62].	119-bit	x	x	x
sect239k1	Binary field - GF(2^m)	[SEC 2].	119-bit	x	x	x
sect233r1	Binary field - GF(2^m)	[SEC 2].	116-bit	x	x	x
sect233k1	Binary field - GF(2^m)	[SEC 2].	116-bit	x	x	x
secp224k1	Prime field – GF(p)	[SEC 2].	112-bit	x	x	x
Brainpool P224r1	Prime field – GF(p)	[RFC 5639].	112-bit	x	x	x
Brainpool P224t1	Prime field – GF(p)	[RFC 5639].	112-bit	x	x	x
sect193r2	Binary field - GF(2^m)	[SEC 2].	96-bit	-	x	-
sect193r1	Binary field - GF(2^m)	[SEC 2].	96-bit	-	x	-
X9.62 prime192v3	Prime field – GF(p)	[ANSI X9.62].	96-bit	-	x	-
X9.62 prime192v2	Prime field – GF(p)	[ANSI X9.62].	96-bit	-	x	-
secp192k1	Prime field – GF(p)	[SEC 2].	96-bit	-	x	-
Brainpool P192r1	Prime field – GF(p)	[RFC 5639].	96-bit	-	x	-
Brainpool P192t1	Prime field – GF(p)	[RFC 5639].	96-bit	-	x	-
X9.62 c2pnb191v3	Binary field - GF(2^m)	[ANSI X9.62].	95-bit	-	x	-
X9.62 c2pnb191v2	Binary field - GF(2^m)	[ANSI X9.62].	95-bit	-	x	-
X9.62 c2pnb191v1	Binary field - GF(2^m)	[ANSI X9.62].	95-bit	-	x	-
X9.62 c2pnb163v1	Binary field - GF(2^m)	[ANSI X9.62].	81-bit	-	x	-
X9.62 c2pnb163v2	Binary field - GF(2^m)	[ANSI X9.62].	81-bit	-	x	-
X9.62 c2pnb163v3	Binary field - GF(2^m)	[ANSI X9.62].	81-bit	-	x	-
sect163r2	Binary field - GF(2^m)	[SEC 2].	81-bit	-	x	-
sect163r1	Binary field - GF(2^m)	[SEC 2].	81-bit	-	x	-
sect163k1	Binary field - GF(2^m)	[SEC 2].	81-bit	-	x	-
Brainpool P160r1	Prime field – GF(p)	[RFC 5639].	80-bit	-	x	-
Brainpool P160t1	Prime field – GF(p)	[RFC 5639].	80-bit	-	x	-
secp160r2	Prime field – GF(p)	[SEC 2].	80-bit	-	x	-
secp160r1	Prime field – GF(p)	[SEC 2].	80-bit	-	x	-
secp160k1	Prime field – GF(p)	[SEC 2].	80-bit	-	x	-

Table 2-4 Non-approved algorithms allowed in the approved mode of operation

Algorithm	Caveat	Use / Function
Key Transport		
KTS (AES Cert #C1707)	Key unwrapping; key establishment methodology provides between 128 and 256 bits of encryption strength. Uses allowances in [FIPS 140-3 IG] D.G, Key transport methods, for key unwrapping using un-authenticated modes of encryption listed on Cert #C1707 without use of an additional approved hash function.	Clone partition objects between partitions, Clone SMK between partitions, Import secret or private key using key wrapping.
KTS (Triple-DES Cert #C1707)	Key unwrapping; key establishment methodology provides 112 bits of encryption strength. Uses allowances in [FIPS 140-3 IG] D.G, Key transport methods, for key unwrapping using un-authenticated modes of encryption listed on Cert #C1707 without use of an additional approved hash function.	Import secret or private key using key wrapping.
KAS-ECC-SSC (Cert #A480)	Key establishment methodology provides between 112 and 256-bits of encryption strength. Curves Non-NIST (as per [FIPS 140-3 IG] C.A): see Table 2-3 above.	Derive key from existing partition secret or private key object. Algorithm implementation used when **HSM Policy (43) Enable low-level math acceleration** is **enabled.**
KAS-ECC-SSC (Cert #A478)	Key establishment methodology provides between 112 and 256-bits of encryption strength. Curves Non-NIST (as per [FIPS 140-3 IG] C.A): see Table 2-3 above for curves over a binary field – GF(2^m) exclusively.	Derive key from existing partition secret or private key object. As per services mapped to KAS-ECC-SSC for Cert #A480 above but where this implementation is only exercised for the listed curves and when **HSM Policy (43) Enable low-level math acceleration** is **disabled**.

Table 2-5: Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed

Algorithm	Caveat	Use / Function
N/A	N/A	N/A

2.5 Non-Approved Algorithms

Non-Approved security functions are not available for use when the module has been configured to operate in the approved mode (see section 13.2).

The following table lists non-approved algorithms supported for use with certain user consumable services when the module is configured in the non-Approved mode of operation during secure initialization.

> **NOTE** The module is capable of supporting a single mode of operation. Transition from an approved to non-approved mode of operation automatically triggers HSM zeroize or decommission module service.

Table 2-6: Non-approved algorithms not allowed in the approved mode of operation.

Algorithm / Function	Use / Function
Symmetric Encryption / Decryption	
ARIA	Perform decrypt operation on user supplied data object, Perform encrypt operation on user supplied data object, Derive key from existing partition secret or private key object, Import secret or private key using key wrapping
CAST3	
CAST5	
DES	
RC2	
RC4	
RC5	
RSA (non-compliant with less than 112 bits of encryption strength)	
RSA X.509[9]	
SEED	
SM4	
Triple-DES (non-compliant for encrypt operations)	
XOR[10]	
Hashing	
HAS-160	Derive key from existing partition secret or private key object, Validate signature or MAC over user supplied data, Perform digest operation on user supplied data
KECCAK	
MD2	
MD5	
RIPEMD-160	
SM3	
Message Authentication Code	
ARIA-CMAC	
SEED-CMAC	

[9] this algorithm allows RSA encryption of a supplied data object without the use of padding. Any required padding is added by the operator ahead of supplying the data to this variant of the RSA encrypt/decrypt function.

[10] this algorithm allows the operator to XOR supplied data with either a supplied base key or key derived from a base key. This function is deprecated for use in any situation where security of the data or key is required.

Algorithm / Function	Use / Function
Triple-DES-CMAC (non-compliant for MAC generation)	Generate signature or MAC over user supplied data, Validate signature or MAC over user supplied data
HMAC (non-compliant with less than 112 bits of encryption strength)	
HAS160-HMAC	
MD5-HMAC	
SM3-HMAC	
RIPEMD160-HMAC	
AES-MAC	
ARIA-MAC	
CAST3-MAC	
CAST5-MAC	
DES-MAC	
RC2-MAC	
RC5-MAC	
SEED-MAC	
SSL3-MD5-MAC	
SSL3-SHA1-MAC	
Triple-DES-MAC	
Triple-DES-x9.19-MAC	
TUAK	
MILENAGE	
COMP128	
Asymmetric	
DSA (non-compliant with less than 112 bits of encryption strength)	Generate signature or MAC over user supplied data, Validate signature or MAC over user supplied data
ECDSA (non-compliant with less than 112 bits of encryption strength)	
EdDSA	
EdDSA PH	
KCDSA	
RSA (non-compliant with less than 112 bits of encryption strength)	
SM2	
SM3	
Key Derivation	
AES[11]	Derive key from existing partition secret or private key object
ARIA	
BIP32	
DES	

[11] AES is non-approved for key derivation when used to derive keys using methods other than as permitted by NIST standard such as [SP800-56Cr2] and [SP800-108r1] in particular, use of AES in ECB or CBC mode directly to derive keys.

Algorithm / Function	Use / Function
MD5	
SHA[12]	
SSL PRE-MASTER	
SSL3-MASTER	
SM3	
Triple-DES	
XOR[13]	
Key Agreement	
ECC (non-compliant with less than 112 bits of encryption strength)	Derive key from existing partition secret or private key object
Diffie-Hellman (key agreement; key establishment methodology; non-compliant with less than 112 bits of encryption strength)	
Key Transport	
AES[14]	Import secret or private key using key wrapping, Export secret or private key using key wrapping, Clone partition objects between partitions[15]
ARIA	
CAST3	
CAST5	
DES	
RC2	
RSA (key wrapping; key establishment methodology; non-compliant with less than 112 bits of encryption strength or when using PKCS#1, v1.5 padding)	
RSA[16]	
SEED	
SM4	
TDES	
Asymmetric Key Generation	
Diffie-Hellman (non-compliant with less than 112 bits of encryption strength)	Generate local symmetric or asymmetric key-pair, Generate domain parameters.
ECC (non-compliant with less than 112 bits of encryption strength)	
KCDSA	
RSA (non-compliant with less than 112 bits of encryption strength)	
SM2	
X9.42 Domain Parameter Generation	

[12] SHA1, SHA2 and SHA3 are non-approved for key derivation when they are used to derive keys in a way that is non-compliant with NIST standards such as [SP800-56Cr2], [SP800-108r1], [SP800-132] and [SP800-135r1].

[13] XOR is non-approved for key derivation when selected as a mechanism to combine supplied user data with an existing module stored key.

[14] AES is non-approved for key transport when used to encrypt keys using methods other than as permitted by NIST standards such as [SP800-38F]. In particular, use of un-authenticated modes of AES for encryption without a separate authentication tag (e.g. signature or MAC) is non-approved.

[15] this service uses both [SP800-56Br2] non-compliant RSA encryption for encryption of a nonce followed by AES encryption of key objects in a [SP800-38F] non-compliant manner when Cloning Protocol Version 1 is used for key export. Later versions of this protocol as separately mapped to this service use approved cryptography.

[16] non-compliant when used for key transport using RSA variants that are [SP800-56Br2] non-compliant.

3 Cryptographic Module Interfaces

3.1 Ports and Interface Overview

The following two figures identify the physical interfaces to the cryptographic module:

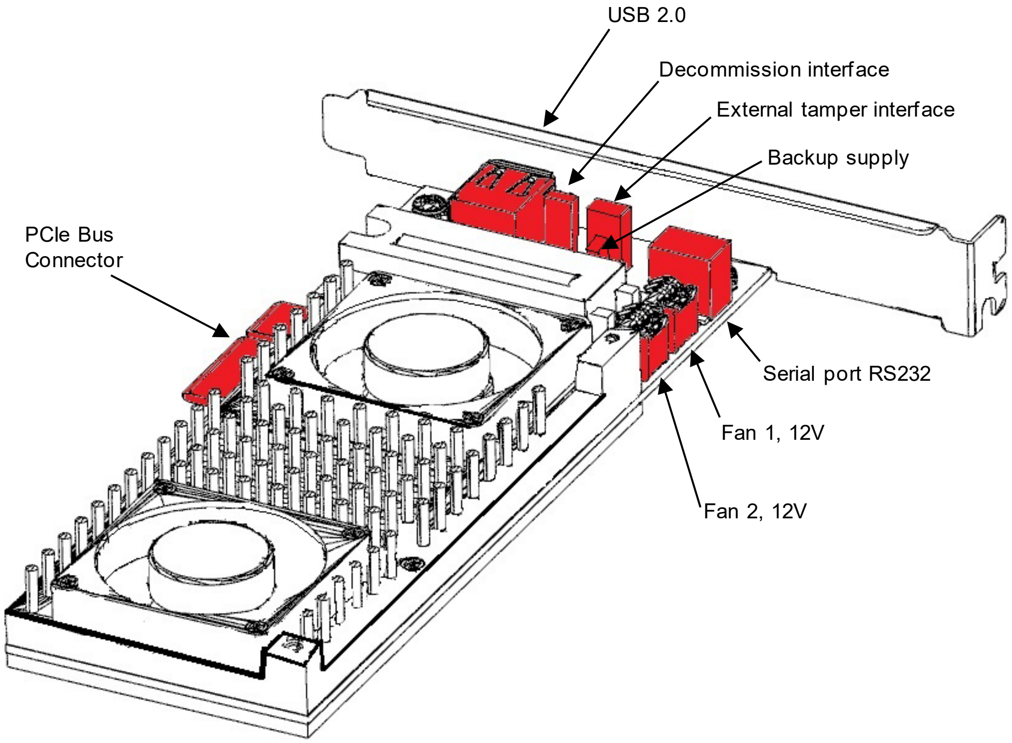

Figure 3-1: Thales Luna K7 Cryptographic Module physical interfaces (non-LED).

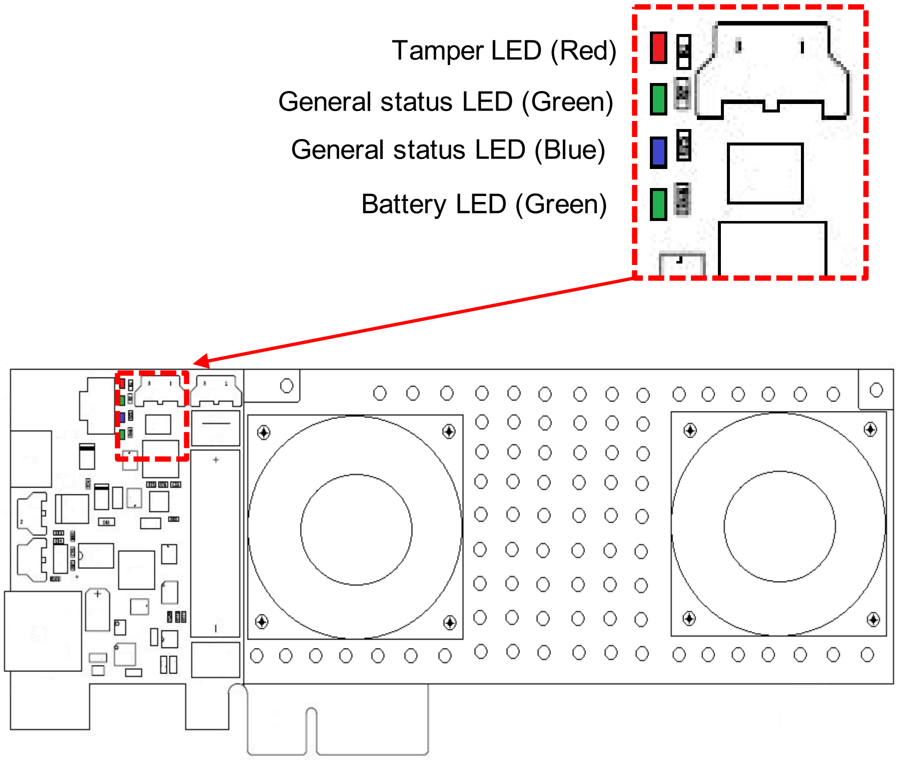

Figure 3-2: Thales Luna K7 Cryptographic Module status LED interfaces.

Where physical interfaces identified in Figure 3-1 and Figure 3-2 are outside the defined cryptographic boundary shown in Figure 2-1 and Figure 2-2, the physical port as listed in the following table corresponds to the internal tracking on the PCB from the physical interface to the module at the point it crossed the cryptographic module boundary.

The following table maps the physical interface to logical interfaces and supported data:

Table 3-1: Ports and interfaces.

Physical port	Logical interfaces	Data that passes over port/interface
USB 2.0, 3.3v[17]	Data input interface, data output interface, control input interface, status output interface.	Encrypted channel for user authentication data entered using the Thales Luna PED alongside the USB-to-Serial dongle used to provide serial access to FM.
Serial port RS232, 3-5.5V, up to 1Mbps	Data input interface, status output interface.	Communication channel to the bootloader when put in ‘interactive mode’ during startup. Boot sequence output during normal startup.

[17] Signals crossing the module boundary are the raw USB 2.0 data packets prior to the addition of physical layer signalling (PHY) and voltage level shifting.

Physical port	Logical interfaces	Data that passes over port/interface
PCIe - Interface PCIe x4 gen3.	Data input interface, data output interface, control input interface, status output interface.	Diagnostics information when the main firmware is operational. Primary interface for user interaction with the module using the ICD protocol as maps to PKCS #11. Encrypted channel for Thales Luna PED when connected remotely. Encrypted channel with Thales Luna client when using the Secure Trusted Channel (STC) (client-to-partition) tunnel.
External Event Interface.	Control input interface.	Control signal used to trigger tamper event.
Decommission Interface	Control input interface.	Control signal used to trigger tamper event.
Tamper LED (Red)	Status output interface.	LED is lit following a tamper event being detected and ahead of module being restarted.
Battery LED (Green)	Status output interface.	LED is lit when the backup 3.6V supply is successful attached to the backup supply header.
General status LED (Green)	Status output interface.	**During boot sequence:** > used to signal progress during the boot process, LED is toggled for each stage in the boot sequence that is successfully achieved; and > when the bootloader is put into interactive mode – the LED is toggled once a second to indicate it is waiting for commands. **General operation:** > LED is lit when main firmware is operating normally; > LED is extinguished should the mode enter an error state; and > LED is extinguished following a factor reset event.
General Status LED (Blue)	Status output interface.	**During boot sequence:** > Turned on when boot sequence starts; and > Turned off following successful validation of the signature on the target main firmware to be loaded. **General operation:** > Off when the main firmware is operating normally; and > Turned on should a ‘halt’ event occur. LED will persist until either a soft or hard reset of the Thales Luna K7 Cryptographic Module is performed.
Power 5V	Power interface	N/A.
Power, 1.8V	Power interface.	N/A.

Physical port	Logical interfaces	Data that passes over port/interface
Power, 3.6V	Power interface	N/A.
Fan 1, 12V	Power interface	N/A.
Fan 2, 12V	Power interface	N/A.

> **NOTE** Interactive mode is a state for the module bootloader where rather than following through a sequence of steps that lead directly to execution of the main firmware without operator intervention, the bootloader can be halted during boot and enters into a state where for a limited period of time, it will offer a number of services in response to commands it receives on the modules serial port.
>
> Services supported are as covered under the ‘bootloader services’ section of Table 4-2.

4 Roles, Services, and Authentication

4.1 Roles

The Thales Luna K7 Cryptographic Module supports the following roles:

Table 4-1: Thales Luna K7 Cryptographic Module Roles

Roles	Principal Duties
HSM Security Officer (HSM SO) [Admin Partition Role]	The HSM SO is responsible for managing the HSM. As such, the HSM SO is authorized to install and configure the HSM, set and maintain global HSM security policies and install FMs. He/she is also able to request the load of new HSM firmware update files (FUF) and new Configuration Update Files (CUF). The HSM SO is able to create and delete partitions, but is not authorized to generate, load or use keys stored on the user partitions that have been created. The HSM SO is able to create, manage and use keys created in the Admin Partition alongside is responsible for initializing the ‘Administrator role’. The HSM SO can reset the Administrator password (configuration dependent). The HSM can have only one HSM SO.
Administrator [Admin Partition Role]	The Administrator is authorized to create, use, transfer and destroy key objects contained in the Admin partition. This role has privileges that are a subset of the HSM SO role.
Partition Security Officer (Partition SO) [User Partition Role]	The Partition SO creates the partition level Partition CO role, sets and changes partition-level policies. This role also has an option to reset the Partition CO password (configuration dependent) following lockout.
Partition Crypto Officer (Partition CO) [User Partition Role]	The Partition CO role is authorized to create, use, destroy and transfer key objects for a given partition. The Partition CO can optionally create the Partition LCO and Partition CU, and perform initial assignment of key authorization data.
Partition Limited Crypto Officer (Partition LCO) [User Partition Role]	The Partition LCO is an optional partition role authorized to create and use key objects, and perform initial assignment of key authorization data. The role is only permitted to delete key objects where per-key authorization is used and the correct authorization data for a given key object can be presented to the cryptographic module.
Partition Crypto User (Partition CU) [User Partition Role]	The Partition CU is the partition role authorized to use the key objects within the partition (e.g. sign, encrypt/decrypt).
Audit User (AU) [Admin Partition Role]	The AU initializes the secret key used to generate Message Authentication Code (MAC) for secure audit messages alongside configuring logging levels for the HSM.

Roles	Principal Duties
Public User [Admin or User Partition Role]	Unauthenticated user with limited access to perform signature verification with public keys where CKA_PRIVATE = false, initialization of the module and roles and to read module status.

> **NOTE** All methods of authentication supported by the module (memorized secret or multi-factor crypto device + memorized secret, incorporate the use of an ID alongside the presentation of the authentication data and are identity based.

Guidance on assuming a given role is covered in section 13.5, 'Assuming Roles'.

The mapping of the cryptographic module's roles services can be found in the table below. In this table, 'Any role' in the 'role' column signifies that any role identified in Table 4-1: Thales Luna K7 Cryptographic Module Roles can access the corresponding service. This includes the 'public user' that is an implicit role and unauthenticated by the module.

Table 4-2: Roles, Service Command, Input and Output.

Role	Service	Input	Output
HSM Management			
Any role.	HSM Factory Reset.	-	-
Any role.	Initialize the HSM.	session, user ID, label, domain, authentication data (if password authentication).	authentication data (if PED authentication), return code.
HSM SO.	Create a user partition.	session, label.	return code.
HSM SO.	Delete a user partition.	session.	return code.
Any role.	Query HSM status.	status information type.	status data, return code.
Any role.	Query partition status.	status information type.	status data, return code.
Any role.	Query HSM configuration.	hsm policy number.	policy status.
Any role.	Query partition configuration.	partition policy numbers.	policy status.
HSM SO.	Set HSM policy.	hsm policy number, value.	return code.
HSM SO (admin partition). Partition SO (user partition).	Set partition policy	partition policy number, value.	return code.
HSM SO.	Update firmware.	session, signed firmware image.	return code.
Any role.	Protect object integrity.	object handle.	Return code.
Any role.	HSM zeroize or decommission.	session.	return code.
Any role.	Trigger user partition zeroize.	session.	return code.
HSM SO.	Load configuration update file.	session, signed configuration update image.	return code.
Any role.	Query the audit log status.	session.	audit log status, return code.

Role	Service	Input	Output
Any role.	Generate secure log record.	session and app_ID, message to log, message type.	return code.
Any role.	Submit external messages for entry into secure audit log.	session, message to be logged.	return code.
AU.	Configure the audit log.	Session, log configuration parameter and value.	return code.
AU.	Export/import audit log secret key.	session, wrapped log secret (import only).	wrapped log secret (export only), return code.
HSM SO, AU.	Set time on HSM real time clock.	session, time.	return code.
AU.	Validate the audit log.	session, audit log segment, audit log key ID.	return code.
Any role.	Request partition STC identity.	session	partition STC identity certificate, return code.
HSM SO (admin or un-initialized user partition). Partition SO (user partition).	Manage STC client	session, client identity certificate.	return code.
HSM SO (admin or un-initialized user partition). Partition SO (user partition).	Query STC status	session, client identity.	client status, return code.
Any role.	Initiate STC tunnel.	-	return code.
Any role.	Send commands to partition with STC tunnel initiated.	command data to tunnel.	command response from module.
HSM SO, Partition CO.	Clone SMK between partitions.	session, SMK ID.	return code.
HSM SO, Administrator, Partition CO, Partition LCO, Partition CU	Clone partition objects between partitions.	session, object ID	return code.
HSM SO, Partition CO.	Rollover SMK for a given partition.	session, SMK ID	return code.
Any role.	Enable/disable STM.	verification data (disable).	verification data (enable), calculated fingerprint (disabled), return code.
Any role.	Clear tamper event.	session.	return code.
Any role.	Request HSM self-test.	session, self-test ID.	return code.
Role Management			
Any role.	Query role status.	session, role.	role status, return code.

Role	Service	Input	Output
HSM SO (required to initialize Administrator). HSM SO, Administrator, AU, Partition SO, Partition CO, Partition LCO, Partition CU, Public User (required to initialize HSM SO, AU or Partition SO). Partition SO (required to initialize Partition LCO or Partition CU).	Initialize role.	session, user ID, role ID, authentication data (if password authentication).	authentication data (if PED configuration), return code.
HSM SO (required to change HSM SO). AU (required to change AU). HSM SO or Administrator (required to change Administrator). Partition SO (required to change Partition SO and Partition CO) Partition CO or Partition LCO (required to change Partition LCO). Note: Roles are not changed, only the role authentication data.	Change authentication data.	session, user ID, role ID, authentication data.	authentication data (if PED configuration), return code.
HSM SO, Administrator, AU, Partition SO, Partition CO, Partition LCO, Partition CU.	Configure partition for high-available recovery / login.	session, HA Login key handle.	return code.
HSM SO, Administrator, AU, Partition SO, Partition CO, Partition LCO, Partition CU.	Login as role.	session, role ID, authentication data.	return code.
Any role.	Close authenticated sessions.	-	return code.
Luna PED Configuration			
HSM SO.	Initialize Remote PED Vector (RPV).	-	return code.
Any role.	Setup Local PED Session.	-	return code.
Any role.	Setup Remote PED Session.	PED_ID.	return code.
Any role.	Send or receive data over PED tunnel (local PED).	When receiving data: encrypted payload. When sending data: Plaintext data to encrypt to PED.	When receiving data: plaintext payload, return code. When sending data: Encrypted data, return code.

Role	Service	Input	Output
Any role.	Send or receive data over PED tunnel (remote PED).	When receiving data: encrypted payload. When sending data: Plaintext data to encrypt to PED.	When receiving data: plaintext payload, return code. When sending data: Encrypted data, return code.
Key Management Activities			
HSM SO, Administrator, Partition CO, Partition LCO.	Generate local symmetric or asymmetric key-pair.	session, generation algorithm, algorithm parameters, public key attributes, private key attributes.	public key handle, private key handle, return code.
Any role.	Generate domain parameters.	session, generation algorithm, algorithm parameters.	domain object handle, return code.
HSM SO, Administrator, Partition CO, Partition LCO.	Derive key from existing partition secret or private key object.	session, algorithm, algorithm parameters, key handles for input derivation keys.	key handle for resulting key, return code.
Any role.	Import public key, certificate, domain object or data objects.	session, object for import.	imported object handle, return code.
HSM SO, Administrator, Partition CO, Partition LCO.	Import secret or private key using key wrapping.	session, unwrapping algorithm, algorithm parameters, handle of wrapping key (asymmetric), handle of key to be unwrapped.	unwrapped key handle, return code.
HSM SO, Administrator, Partition CO, Partition LCO.	Export secret or private key using key wrapping.	session, wrapping algorithm, algorithm parameters, handle of wrapping key, handle of key to be wrapped.	wrapped key, return code.
Any role.	Read non-sensitive key attribute where **CKA_PRIVATE = false** for a given key object.	session, object attributes.	object data, return code.
HSM SO, Administrator, Partition CO, Partition LCO, Partition CU, Public User.	Read non-sensitive key attribute where **CKA_PRIVATE = true** for a given key object.	session, object attributes.	object data, return code.
HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	Insert key from external storage using SKS.	session, SKS key blob.	inserted key object handle, return code.
HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	Extract key to external storage using SKS.	session, key handle.	SKS key blob, return code.
Cryptographic Services			
HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU.	Re-seed partition DRBG.	session, seed.	return code.

Role	Service	Input	Output
HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU.	Extract entropy from partition DRBG.	session, size of random data requested.	random data, return code.
HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU.	Perform digest operation on user supplied data.	session, data to hash.	hash result, return code.
Any role.	Perform encrypt operation on user supplied data object.	session, algorithm, algorithm parameters, data to encrypt.	encrypted data, return code.
HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	Perform decrypt operation on user supplied data object.	session, algorithm, algorithm parameters, data to decrypt.	decrypted data, return code.
HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	Generate signature or MAC over user supplied data.	session, algorithm, algorithm parameters, data to sign.	signature, return code.
HSM SO, Administrator, Partition SO[18], Partition CO, Partition LCO, Partition CU.	Validate signature or MAC over user supplied data.	session, algorithm, algorithm parameters, data to verify, signature.	return code.
Bootloader Services			
Any role.	Set/read scratchpad flag to signal to main firmware.	flag, value	return code.
Any role.	Request complete erase of the HSM main firmware image, loaded FM and key stores (excludes erase of bootloader).	-	-
Any role.	Read Vital Product Data programmed at manufacture.	-	VPD contents.
Any role.	Request authentication and execution of main firmware.	-	-
Functionality Module Management			
HSM SO.	Download FM.	signed binary for FM.	return code.
HSM SO, Administrator.	Activate SMFS.	-	return code.
Any role.	Store/retrieve data from SMFS.	data.	return code.
Any role.	Delete FM.	FM ID	return code.
Any role.	Delete SMFS.	-	return code.
Any role.	Get FM status.	FM ID	FM status, return code.

[18] Partition SO is able to validate signatures using public keys where `CKA_PRIVATE` is set to `0`.

4.2 Roles and Authentication

4.2.1 Authentication Mechanism Summary

All roles, except for the Public User, must authenticate to the module by providing their authentication data.

If configured with PED, all roles must authenticate using an iKey. When a role is initialized, a module generates the authentication data as a 48-byte random value and writes it to a iKey. Optionally, the Crypto-Officer, Limited Crypto Officer and Crypto-User roles can be configured to use two-factor authentication by also assigning a password to the role.

If configured with Password, all roles must authenticate using a minimum of an 8-character password. When a role is initialized under this configuration, the operator enters the initial password for the role.

Regardless of configuration (PED or Password), the password is delivered to the module encrypted with the public key from the Password Encryption Certificate (PEC) using KTS-OAEP-basic from [SP800-56Br2].

Table 4-3: Roles and Authentication

Role	Authentication Method [SP800-140E]		Authentication Strength
	Password Configuration	PED Configuration	
HSM SO	Memorized Secret.	Multi-Factor Crypto Device.	iKey: 48-byte random authentication data generated when a role is initialized and stored on iKey. The probability of guessing the authentication data in a single attempt is 1 in 2^{384}. With a maximum of 6000 failed login attempts per minute. User provided byte array (minimum 8 bytes): Memorized secret when set using tools supplied with the module are limited to a character set of 86 characters[19] presented to the module as their ASCII byte representation. The strength of an 8 character password with character set size of 86 is $\log_2(86^8)$. This makes the probability of guessing the memorized secret in a single attempt 1 in 2^{51} The module supports a maximum of 6000 failed login attempts per minute when the failed login count for a given role is disabled. Automatic lock-out: This feature, which is enabled by default, can be used to limit the impact of brute force attacks on login and is covered in more detail in section 4.2.3.
Auditor	Memorized Secret.	Multi-Factor Crypto Device.	
Partition SO	Memorized Secret.	Multi-Factor Crypto Device.	
Partition CO	Memorized Secret.	Multi-Factor Crypto Device + optional Memorized Secret.	
Partition LCO	Memorized Secret.	Multi-Factor Crypto Device + optional Memorized Secret.	
Partition CU	Memorized Secret.	Multi-Factor Crypto Device + optional Memorized Secret.	
Administrator	Memorized Secret.	Multi-Factor Crypto Device.	
Public User	Not Required.	N/A.	N/A.

When using the password authentication mechanism, the module encrypts a known check-word under a key derived using PBKDF from [SP800-132] and option 1a from section 5.4, 'Using the Derived Master Key to Protect Data'. During a login attempt, the module generates a key from the supplied password, and attempts to decrypt a known checkword. Successful login is achieved if the decrypted checkword matches the expected value. If successful, the PBKDF derived key is used to remove a layer of encryption from the module stored User Storage Key (USK)[20].

[19] Supported characters by tools used with the module to configure memorized secret are limited to:
`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  !@#$%^*()-_=+[]{}/:',.~`

[20] When 'decommission' is enabled as a module capability, the USK is independently encrypted in storage under a 256-bit module generated AES key.

The length of the password used as input to the PBKDF function is consistent with the password length selected by the authenticating user, which is required to be between 7 and 255 characters long. Where passwords are randomly generated, the probability of successfully guessing the password and deriving the storage key for a minimum password length of 7 characters is 1 in 2^{56}. This probability is significantly reduced if random passwords are not used.

Guidance in Appendix A, 'Security Considerations' of [SP800-132] should be consulted when picking an appropriate password length in situations where encryption layers derived from the user password are required to protect the confidentiality of module protected user keys.

The module uses an iteration count of 1000 when generating the key used to decrypt the checkword. This limit has been set to account for the fact that objects encrypted under the [SP800-132] derived key are never exported from the module and are exclusively stored inside its cryptographic boundary where they are physically protected. In addition, the module supports lock-out of all identities following a configurable number of failed login attempts where this is the primary mechanism offered by the module to protect against brute-forcing of memorized secret.

4.2.2 Activation

If PED authentication is configured, the Crypto-Officer, Limited Crypto Officer and Crypto-User roles can be configured to use a two-step authentication process. The first stage is termed "Activation" and is performed using a PED key. Once activated, access to key material and cryptographic services is not allowed until the second stage of authentication, 'User Login', has been performed using the role's password.

Once activated, a role stays activated until the role is explicitly deactivated, deleted or the module is reset[21].

4.2.3 Account lockout behaviors

In addition to the cryptographic strength of the authentication mechanisms, all authenticating roles have the ability to maintain a failed authentication count that can be configured to stop attempts to brute force authentication data.

The maximum supported failed authentication attempts can be set to between 3 and 10 for each role with the following lockout behaviors observed:

- lockout of the HSM SO role will trigger the HSM zeroize or decommission service;
- lockout of the Partition SO will trigger the Trigger user partition zeroize service; and
- lockout of the Administrator, AU, Partition CO, Partition LCO, Partition CU roles will block future authentication attempts until the role is unlocked using the Change authentication data service.

4.3 Approved Services

All services listed in the table below can be accessed in approved mode and when in this mode exclusively use the security functions listed in Table 2-2 and

When the module is operating in this mode, security functions in section 2.5 are disabled and blocked from being used.

As notes on the content of Table 4-4:

[21] A module is reset in response to a trigger signal being received on the External Event input, Decommission signal and EFP violations, loss of power or a request from a host application.

- In the ‘Approved Security Functions’ column:
  - ‘Algorithms’ maps the target service to cryptography from standards referenced in [SP800-140Cr1] alongside corresponding CAVP certificates from Table 2-2 or ‘non-Approved but Allowed’ cryptography from Table 2-4;
  - ‘Key Management technique’ maps the target service to cryptography from standards referenced in [SP800-140Dr1] alongside corresponding CAVP certificates from Table 2-2 or ‘non-Approved but Allowed’ cryptography from Table 2-4;
  - ‘Authentication Technique’ lists the permitted authentication mechanism as specified in [SP800-140E];
  - For RSA, ECDSA and AES-KW, where multiple algorithms may be used based on module settings as covered in Table 2-2 only the primary implementation is listed in the table.
- In the ‘Roles Column’:
  - ‘Any role’ – maps to all defined roles for the module. This includes HSM SO, Administrator, AU, Partition SO, Partition CO, Partition LCO, Partition CU and Public User.
- In the ‘Access Rights to Keys and/or SSPs’ column:
  - 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;
  - E = Execute: The module uses the SSP in performing a cryptographic operation; and
  - Z = Zeroize: The module zeroizes the SSP.
- In the ‘Indicator Column’:
  - IND_1 – **Partition Policy (42) Enable CPv1** is set to `false`, **Partition Policy (43) Enable non-FIPS Algorithms** is set to `false`, **HSM Policy (52), Restrict FM Privilege** is set to `true,` **HSM Policy (56), Allow User Defined ECC Curves** is set to `false`, AND return code is `CKR_OK`;
  - IND_2 – **Partition Policy (42) Enable CPv1** is set to `false`, Partition Policy (43) Enable non-FIPS Algorithms is set to `false`, HSM Policy (52), Restrict FM Privilege is set to true, HSM Policy (56), Allow User Defined ECC Curves is set to `false`, AND return code is `PED_RET_OK` or `SP_RET_OK`; and
  - IND_3 – **Partition Policy (42) Enable CPv1** is set to `false`, **Partition Policy (43) Enable non-FIPS Algorithms** is set to `false`, **HSM Policy (52), Restrict FM Privilege** is set to `true,` **HSM Policy (56), Allow User Defined ECC Curves** is set to `false`, AND return code is `0`.

> **NOTE** While default setting for **HSM Policy (52), Restrict FM Privilege** is `false,` this setting is only relevant to a FIPS approved configuration following load of the ‘Enable Functionality Module CUF’. This unlocks the ability to enable the related ‘**HSM Policy (50) Allow Functionality Modules**’. Prior to HSM Policy (50) being set to `true`, the setting of **HSM Policy (52) Restrict FM Privilege** is redundant to the modules operation as functionality modules can’t be loaded and the policy only relates to commands received by the module from within code executing within a sandboxed functionality module.

- In the ‘Keys and/or SSPs’ column:
  - For a complete description of SSP referenced from the table, see Table 9-1.

Table 4-4: Module Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
HSM Management						
HSM Factory Reset.	Factory reset deletes all roles (including HSM SO), all users and objects and sets all HSM settings and policy to values defined in pre-loaded configuration update files.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	PSK, USK, DRBG Key, DRBG Seed, DRBG V, KCV, SMK, STC-PID$_{PUB}$, STC-PID$_{PRIV}$, STC-CID$_{PUB}$, STC-PID$_{PUB}$, STC-PKA$_{PUB}$, STC-PKA$_{PRIV}$, STC-CKA$_{PUB}$, STC-PEN, STC-PMA, STC-PIV, STC-CEN, STC-CMA, STC-CIV, HA$_{PUB}$, HA$_{PK}$, RND, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), DRBG Key, SALK, CWK$_{HSM}$, CWK$_{PED}$, DEK$_{HSM}$, DMK$_{HSM}$, DEK$_{PED}$, DMK$_{PED}$, AEK, AEK-EK, AccessID.	Any role.	**Z:** (for ALL partition) PSK, USK, DRBG Key, DRBG Seed, DRBG V, KCV, SMK, STC-PID$_{PUB}$, STC-PID$_{PRIV}$, STC-CID$_{PUB}$, STC-PID$_{PUB}$, STC-PKA$_{PUB}$, STC-PKA$_{PRIV}$, STC-CKA$_{PUB}$, STC-PEN, STC-PMA, STC-PIV, STC-CEN, STC-CMA, STC-CIV. HA$_{PUB}$, HA$_{PK}$, RND, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys). In addition, the following HSM level keys are erased: DRBG Key, SALK, CWK$_{HSM}$, CWK$_{PED}$, DEK$_{HSM}$, DMK$_{HSM}$, DEK$_{PED}$, DMK$_{PED}$. **E:** AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Initialize the HSM.	This service is used to initialize the HSM on first use or following zeroization. Actions performed by this service include: > resets the admin partition; > deletes all user partitions; > initializes the HSM SO role; > creates / selects KCV to be used with the admin partition; > generates PEC, PEK, USK and PSK keys for the admin partition; and > encrypts keys for storage.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** ESV (Cert #98), CKG, CTR_DRBG (Cert #C1707), SHA #C1707) – SHA2-512, (PBKDF (Cert #A480), KBKDF (Cert #C1707). **Authentication technique:** N/A.	For ALL partition if present – USK, PSK, SMK, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys). For user partitions: HA$_{PUB,}$ HA$_{PK}$ (for all roles). For admin partition - HA$_{PUB}$, HA$_{PK}$ for Admin role only. PEK, PEC, USK, PSK, DRBG Key, DRBG Seed, DRBG V, KCV. PSK, USK, GSK, User Password, Password, PED Authentication Data, Stored User Password Hash, AEK, AEK-EK, AccessID.	Any role.	**Z:** For ALL partition if present – USK, PSK, SMK, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys). For User Partitions –HA$_{PUB,}$ HA$_{PK}$ (for all roles). For Admin Partition –HA$_{PUB}$, HA$_{PK}$ for Admin role only. **G and W:** PEK, PEC, USK, PSK, DRBG Key, DRBG Seed, DRBG V, KCV, Stored User Password Hash. **G:** PED Authentication Data, User Password. **E:** User Password, PED Authentication Data, Password, PSK, USK, GSK, AEK, AEK-EK, AccessID.	IND_1.
Create a user partition.	This service creates a user partition at the request of the HSM SO. The user partition is created in memory but roles associated with the partition are retained in an un-initialized state.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** ENT (P), CKG, SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707). **Authentication technique:** N/A.	DRBG Key, DRBG Seed, DRBG V, KCV, AEK, AEK-EK, AccessID.	HSM SO.	**W:** DRBG Key, DRBG Seed, DRBG V, KCV. **E:** AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Delete a user partition.	This service is used to delete an existing user partition. During deletion, the module zeroizes all objects associated with the partition.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	PSK, USK, DRBG Key, DRBG Seed, DRBG V, KCV, SMK, STC-PID$_{PUB}$, STC-PID$_{PRIV}$, STC-CID$_{PUB}$, AEK, AEK-EK, AccessID, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys) .	HSM SO.	**Z:** PSK, USK, DRBG Key, DRBG Seed, DRBG V, KCV, SMK, STC-PID$_{PUB}$, STC-PID$_{PRIV}$, STC-CID$_{PUB}$, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys).	IND_1.
Query HSM status.	This service is used to retrieve general status information on the module including items such as: > hardware, bootloader and main firmware versions; > module serial number; > module state (e.g. tampered, zeroized, initialized); > authenticated roles for active session (if present); > number of configured partitions; and > general error messages and logs.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Query partition status.	This service is used to retrieve general status information on a target partition including items such as: > partition label and serial number; > partition state (e.g. token initialized, user initialized, login required); > RPV and KCV state and active SMK ID. > number of stored objects; and > used and free storage space.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.
Query HSM configuration.	This service is used to retrieve information on HSM configuration and policy settings.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.
Query partition configuration.	This service is used to retrieve information on the configuration and policy settings for a target partition.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Set HSM policy.	This service is used to set available HSM policy settings. HSM policy can only be configured if the corresponding configuration item is enabled which is defined based on loaded configuration update files. If a given policy being set is a 'destructive policy' – changing the setting will trigger zeroization of the module.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), USK, PSK, KCV, SMK, STC-PID$_{\text{PRIV}}$, STC-PID$_{\text{PUB}}$, HOK, HOC, ECC HOK, ECC HOC,TUK4,TWC4, CITS-DAK, CITS-DAC, ECC DAK, ECC DAC, HSM-SKA-CREMOTE, HSM-SKA-KLOCAL, PAC, RPV-C, PEK, PEC, AEK, AEK-EK, AccessID..	HSM SO.	**Z: for all destructive policies** – Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), USK, PSK, KCV, SMK, STC-PID$_{\text{PRIV}}$, STC-PID$_{\text{PUB}}$. The following are erased in addition if **HSM Policy (50) Allow Functionality Modules** is **enabled**: HOK, HOC, ECC HOK, ECC HOC,TUK4,TWC4, CITS-DAK, CITS-DAC, ECC DAK, ECC DAC, HSM-SKA-CREMOTE, HSM-SKA-KLOCAL, PAC, RPV-C, PEK, PEC.	IND_1.
Set partition policy	This service is used to set available partition policy settings. Partition policy can only be configured if dependencies at the HSM level of configurations and policy are met. If a given policy being set is a destructive policy – changing the setting will trigger zeroization of all user objects stored in the admin partition.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), SMK, AEK, AEK-EK, AccessID.	HSM SO (admin partition). Partition SO (user partition).	**Z:** Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), SMK. **E:** AEK, AEK-EK, AccessID.	IND_1.
Update firmware.	This service validates and then loads a new main firmware image (excluding bootloader). The replacement image is signed using RSA PKCS #1-v1.5 signature using SHA2-384 and 4096-bit modulus.	**Algorithms:** RSA (Cert #A480) – Signature Verification, SHA (Cert #C1707). **Key management technique:** N/A. **Authentication technique:** N/A.	Root Certificate, Firmware Signing Certificate, AEK, AEK-EK, AccessID.	HSM SO.	**E:** Root Certificate, Firmware Signing Certificate, AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Protect object integrity	This is an internal module service used to protect the integrity of all stored object and configuration data. All objects are stored with a SHA2-256 hash, which is checked on retrieval ahead of object use.	**Algorithms:** SHA (Cert #C1707). **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.
HSM zeroize or decommission.	This service zeroizes the module with the exception of the following: > SSP associated with the Audit partition and AU role are not zeroized; > RPV persists allowing use of remote PED during re-initialization.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	<As per service 'HSM factory reset' above but excluding: SALK, RPV.>	Any role.	**Z:** <As per service 'HSM Factory reset' above but excluding: SALK, RPV.>	IND_1.
Trigger user partition zeroize.	This service erases of keys stored in a user partition and resets Any role to their un-initialized state.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	USK, PSK, KCV, SMK, STC-PID$_{PUB}$, STC-PID$_{PRIV}$, STC-CID$_{PUB}$, STC-PID$_{PUB}$, STC-PKA$_{PUB}$, STC-PKA$_{PRIV}$, STC-CKA$_{PUB}$, STC-PEN, STC-PMA, STC-PIV, STC-CEN, STC-CMA, STC-CIV, HA$_{PUB}$, HA$_{PK}$, RND, , AEK, AEK-EK, AccessID, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys).	Any role.	**Z:** USK, PSK, KCV, SMK, STC-PID$_{PUB}$, STC-PID$_{PRIV}$, STC-CID$_{PUB}$, STC-PID$_{PUB}$, STC-PKA$_{PUB}$, STC-PKA$_{PRIV}$, STC-CKA$_{PUB}$, STC-PEN, STC-PMA, STC-PIV, STC-CEN, STC-CMA, STC-CIV, HA$_{PUB}$, HA$_{PK}$, RND, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys). **E:** AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Load configuration update file.	This service validated the signature on a loaded configuration update file ahead of its contents being stored on the module. The configuration update file defines the default settings for one or a number of HSM or Partition level configuration and policy settings. Configuration update files are signed using RSA PKCS #1-v1.5 signature using SHA2-384 and 4096-bit modulus.	**Algorithms:** RSA (Cert #A480) – Signature Verification, SHA (Cert #C1707). **Key management technique:** N/A. **Authentication technique:** N/A.	Root Certificate and License Signing Certificate, AEK, AEK-EK, AccessID.	HSM SO.	**E:** Root Certificate, License Signing Certificate, AEK, AEK-EK, AccessID.	IND_1.
Query the audit log status.	This service is used to retrieve general status information on the secure audit log.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	AEK, AEK-EK, AccessID.	IND_1.
Generate secure log record	This is an internal module service to the cryptographic module used to add records to the secure audit log. AccessID are hashed with SHA2-512 ahead of inclusion in the log. Records are given a MAC using HMAC-SHA2-256.	**Algorithms:** HMAC (Cert #C1718) – HMAC-SHA2-256, SHA (Cert #C1718) – SHA2-256, SHA2-512. **Key management technique :** N/A. **Authentication technique :** N/A.	SALK, AEK, AEK-EK, AccessID.	Any role.	**E:** SALK will be used if AU role initialised**,** AEK, AEK-EK, AccessID.	IND_1.
Submit external messages for entry into secure audit log.	The service is used by processes running outside the boundary of the module to submit entries to the module audit log. Entries are identified in the log as having come from an external source.	**Algorithms:** HMAC (Cert #C1718) – HMAC-SHA2-256, SHA (Cert #C1718) – SHA2-256, SHA2-512. **Key management technique :** N/A. **Authentication technique :** N/A.	SALK, AEK, AEK-EK, AccessID.	Any role.	**E:** SALK will be used if AU role initialised**,** AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Configure the audit log.	This service is used to configure which audit events are to be recorded in the secure audit log and in addition to configure the location of the secure logging daemon used to extract log sections from the module. Events are selected based on logging categories assigned to different services with some events always logged unconditionally (e.g. tamper events and self-test failures).	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	AU.	**E:** AEK, AEK-EK, AccessID.	IND_1.
Export/import audit log secret key.	This service exports or imports and encrypted copy of the SALK. This service can be used to allow validation of the authenticity of extracted log sections between modules.	**Algorithms:** N/A. **Key management technique:** AES (Cert #C1707) – KWP mode with 256-bit key, KBKDF (Cert #C1707). **Authentication technique:** N/A.	RDK, SALK, AEK, AEK-EK, AccessID.	AU.	**E:** RDK, AEK, AEK-EK, AccessID. **R:** SALK	IND_1.
Set time on HSM real time clock.	This service sets the time on the module real time clock as used for time-stamps in the secure audit log alongside enforcing the validity dates on keys.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	HSM SO, AU.	**E:** AEK, AEK-EK, AccessID.	IND_1.
Validate the audit log.	This service checks both the integrity and authenticity of extracted sections of the secure module audit log.	**Algorithms:** HMAC (Cert #C1718) – HMAC-SHA2-256, SHA (Cert #C1718) – SHA2-256. **Key management technique :** N/A. **Authentication technique :** N/A.	SALK, AEK, AEK-EK, AccessID.	AU.	**E:** SALK, AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Request partition STC identity.	This service extracts a copy of the partition identity and returns it in a file to transfer out-of-band to the client. If the target key-pair does not exist, this service generates the corresponding ECDSA key-pair alongside packaging the public component as a signed certificate (signed using the ECC HOK and ECDSA SHA2-384). The private key is stored encrypted under the GSK using AES-256 in KWP mode.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key, ECDSA (Cert #C1707) –Signature Generation using curve P521, SHA (Cert #C1707) – SHA2-384 for certificate signing. **Key management technique:** ESV (Cert #98), CKG, SHA #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), ECDSA (Cert #C1707) – Key Generation with curve P521. **Authentication technique:** N/A.	DRBG Key, DRBG Seed, DRBG V, GSK, STC-PID$_{PUB}$, STC-PID$_{PRIV}$, ECC HOK, AEK, AEK-EK, AccessID.	Any role (Admin partition identity retrieval). HSM SO (uninitialized user partition identity). Partition SO (initialized user partition).	**E:** DRBG Key, DRBG Seed, DRBG V, GSK, STC-PID$_{PUB}$, AEK, AEK-EK, AccessID. **G/W:** STC-PID$_{PRIV}$, STC-PID$_{PUB}$. **W:** DRBG Key, DRBG Seed, DRBG V.	IND_1.
Manage STC client	This service is used to register or de-register a client based on loading its public key onto the module and assigning it to the list of authorized clients for a target partition.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	STC-CID$_{PUB}$, AEK, AEK-EK, AccessID.	HSM SO (admin or un-initialized user partition). Partition SO (user partition).	**Z:** STC-CID$_{PUB}$ (if de-registering client) **E/W:** STC-CID$_{PUB}$ (if registering a client), AEK, AEK-EK, AccessID.	IND_1.
Query STC status.	This service is used to check whether STC is active alongside what cipher-suite is in use for a target client to partition STC tunnel.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	None.	HSM SO (admin or un-initialized user partition). Partition SO (user partition).	None.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Initiate STC tunnel.[22]	This service establishes a pre-configured STC tunnel between a client and the target partition. As a pre-requisite to this service, the client and partition must be pre-registered with each other.	**Algorithms:** AES (Cert #C1707) – AES-256 in GCM or CTR mode, HMAC (Cert #C1707) – HMAC-SHA2-512 (used when AES-256 used in CTR mode), ECDSA (Cert #C1707), SHA (Cert #C1707) – SHA2-512. **Key management technique:** ESV (Cert #98), CKG, SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), ECDSA (Cert #C1707) – Key Generation with curve P521, KAS (Cert #A480) – fullUnified with curve P521 and oneStep KDF with SHA2-512. **Authentication technique:** Single-Factor Crypto Software.	DRBG Key, DRBG Seed, DRBG V, STC-PID$_{PRIV}$, STC-CID$_{PUB}$, ECC HOC, GSK. STC-PKA$_{PUB}$, STC-PKA$_{PRIV}$**,** STC Master Shared Secret, STC-PEN, STC-PMA, STC-PIV, STC-CEN, STC-CMA, STC-CIV.	Any role.	**E:** DRBG Key, DRBG Seed, DRBG V, STC-PID$_{PRIV}$, STC-CID$_{PUB}$, ECC HOC, GSK. **G:** STC-PKA$_{PUB}$, STC-PKA$_{PRIV}$**,** STC Master Shared Secret, STC-PEN, STC-PMA, STC-PIV, STC-CEN, STC-CMA, STC-CIV. **W:** DRBG Key, DRBG Seed, DRBG V.	IND_1.

[22] To initiate and use STC the HSM SO must have done the initial configuration prior to running the command.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Send commands to partition with STC tunnel initiated.	The service is used to tunnel commands from the client to a target partition over an established STC tunnel. Commands are confidentiality, integrity and replay protected while transiting the tunnel.	**Algorithms:** AES (Cert #C1707) – AES-256 in GCM or CTR mode, HMAC (Cert #C1707) – HMAC-SHA2-512 (used when AES-256 used in CTR mode). **Key management technique:** N/A. **Authentication technique:** Single-Factor Crypto Software.	STC-PEN, STC-PMA, STC-PIV, STC-CEN, STC-CMA, STC-CIV.	Any role.	**E:** STC-PEN, STC-PMA, STC-PIV, STC-CEN, STC-CMA, STC-CIV.	IND_1.

Clone SMK between partitions.	This service uses the cloning protocol to establish a shared key between source and destination partitions and then to transfer a selected SMK encrypted under this shared key between partitions. Either CPV3 or CPV4 is supported for both import and export of SMK.	**Algorithms:** CPV4: CTR_DRBG (Cert #C1707), AES (Cert #C1707) – AES-256 in GCM mode or CTR and either HMAC-SHA2-512 or HMAC-SHA3-512 (Cert #C1707). CPV3: CTR_DRBG (Cert #C1707), AES (Cert #C1707) – AES-256 in KWP. **Key management technique:** ESV (Cert #98), CKG, SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), SHA (Cert #C1707) – SHA2-512. CPV4 - KAS (Cert #A480) – Ephemeral Unified and OneStep KDF (with pre-shared 256-bit key as additional input) with either-SHA2-512 or SHA3-512 as the PRF. CPV3 - KAS (Cert #A480) – KAS1-basic using 4096-bit modulus and OneStep KDF (with pre-shared 256-bit key as additional input) with SHA2-512. **Authentication technique:** Single-Factor Crypto Software.	DRBG Key, DRBG Seed, DRBG V, Root Certificate, MIC, HOC (or FM equivalent), KCV and Cloning Transfer Key, SMK, AEK, AEK-EK, AccessID and; CPV3 KEV$_t$, TUK4, TWK4. CPV4 CPV4 Key Agreement Private Key, CPV4 Key Agreement Source Public Key, CPV4 Key Agreement Source Destination Public Key, CPV4 Key Agreement Shared Secret, CPV4 Key Transport Key, CPV4 Session Key, CPV4 Session Salt, CPV4 Per-Blob Salt, CPV4 Per-Blob Encryption Key, CPV4 Per-Blob MAC Key, CPV4 Messaging Private Key, CPV4 Messaging Public Key.	HSM SO is able to clone (or receive) the SMK from/to the Admin Partition. Partition CO is able to clone (or receive) the SMK from/to the user partition.	**G/E:** KEV$_s$, KEV$_t$, CPV4 Key Agreement Private Key, Key Agreement Source Public Key, CPV4 Key Agreement Shared Secret, CPV4 Key Transport Key, CPV4 Session Key, CPV4 Session Salt, CPV4 Per-Blob Salt, CPV4 Per-Blob Encryption Key, CPV4 Per-Blob MAC Key, Cloning Transfer Key. **E:** DRBG Key, DRBG Seed, DRBG V, Root Certificate, MIC, HOC (or FM equivalent), TWC3, TWK3 TUK4, TWK4, CPV4 Messaging Private Key, CPV4 Messaging Public Key, KCV, Key Agreement Destination Public Key, AEK, AEK-EK, AccessID. **R/W**: SMK. **W:** DRBG Key, DRBG Seed, DRBG V.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Clone partition objects between partitions.	This service supports use of CPV4, CPV3 for key object import and export.	<as per service *Clone SMK between partitions*>	<as per service *Clone SMK between partitions*.>	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	<as per service *Clone SMK between partitions*.>	IND_1.
Rollover SMK for a given partition.	This service generates a new SMK and demotes the current SMK. One transfer of all externally stored keys to the new SMK is complete, this service can separately be used to zeroize the old SMK.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** ESV (Cert #98), CKG, SHA #C1707) – SHA2-512, CTR_DRBG (Cert #C1707). **Authentication technique:** N/A.	USK, DRBG Key, DRBG Seed, DRBG V, SMK, DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID.	HSM SO, Partition CO.	**E:** USK, DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID. **W:** SMK, DRBG Key, DRBG Seed, DRBG V.	IND_1.
Enable/disable STM.	This service generates or validates a checksum for full module integrity. This includes integrity of all data stored in memory (includes all keys, executables, configuration data etc). Typically, the feature is used to guarantee storage integrity during shipping or an extended period of storage. The feature generates a seed and checksum that are stored outside the module by the user activating STM.	**Algorithms:** SHA (Cert #C1707) – SHA2-256. **Key management technique:** ESV (Cert #98), SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707). **Authentication technique:** N/A.	STM Nonce, DRBG Key, DRBG Seed, DRBG V (DRBG only used to create STM None on enabling STM), AEK, AEK-EK, AccessID.	**Modules in zeroized State**: Any role. **Initialized Module:** HSM SO.	**G:** STM Nonce. **E:** STM Nonce, DRBG Key, DRBG Seed, DRBG V (DRBG only used to create STM None on enabling STM), AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Clear tamper event.	This service allows module tamper event to be cleared. Following a tamper event, the module will restart and the user is forced to acknowledge the tamper event ahead of the module returning to an operational state. The role required to acknowledge the tamper event depends on the state the tamper event left the module.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	HSM SO (required if the tamper event did not result in zeroization of the module). Any role (in situations where the tamper event triggered a module halt but did not zeroize the HSM).	**E:** AEK, AEK-EK, AccessID.	IND_1.
Request HSM self-test.	This service allows components of the power-on self-test to be triggered on demand. The service supports re-run of the entire power-on self-test alongside selection of individual tests to re-run.	**Algorithms:** <All general algorithms from listed in Table 2-2 and Table 2-4> **Key management technique:** <All Key Establishment and Key Transport methods from listed in Table 2-2 and Table 2-4.> **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.
Role Management						
Query role status.	This service returns status in relation to a target role (e.g. whether the role is initialized or not).	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Initialize role.	This service is used to initialize a role (admin partition or user partition). Privileges required to initialize a role are dependent on the role being initialized.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key, SHA #C1707) – SHA2-256. **Key management technique:** ESV (Cert #98), CKG, SHA (Cert #C1707) – SHA2-512, DRBG (Cert #C1707), PBKDF (Cert #A480), KBKDF (Cert #C1707). **Authentication technique:** N/A.	DRBG Key, DRBG Seed, DRBG V, USK, PSK, KEK, PEK. PEK, PEC, PED Authentication Data, User Password (if challenge-secret enabled), Password, Stored User Password Hash, AEK, AEK-EK, AccessID.	HSM SO (required to initialize Administrator). Any role. (required to initialize HSM SO, AU or Partition SO). Partition SO (required to initialize Partition LCO or Partition CU).	**E:** DRBG Key, DRBG Seed, DRBG V, USK, PSK, KEK, PEK, AEK, AEK-EK, AccessID. **W:** PEK, PEC, PED Authentication Data, Stored User Password Hash (if challenge-secret enabled), Password.	IND_1.
Change authentication data.	This service is used to change authentication data for a given role (admin partition or user partition). Privileges required to change the authentication data are dependent on the target role and HSM configuration.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** PBKDF (Cert #A480), KBKDF (Cert #C1707), CKG. **Authentication technique:** N/A.	USK, PSK, KEK, PEK. PEK, PEC, PED Authentication Data, User Password (if challenge-secret enabled), Password, AEK, AEK-EK, AccessID.	HSM SO (required to change HSM SO). AU (required to change AU). HSM SO or Administrator (required to change Administrator). Partition SO (required to change Partition SO and Partition CO) Partition CO or Partition LCO (required to change Partition LCO).	**R:** USK. **E:** KEK, PEK (if password authentication used), PED Authentication Data, Stored User Password Hash (if challenge-secret enabled), Password, AEK, AEK-EK, AccessID. **W:** USK, Stored User Password Hash.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Configure partition for high-available recovery / login.	This service is used to setup, authorize and use the high-availability recovery feature for partitions. A given role can only configure this feature for the role they are assuming for a given session.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key, RSA (Cert #C1707) – RSA PKCS #1-v1.5 signature validation with SHA2-384 and 4096-bit modulus, SHA (Cert #C1707) – SHA2-256, SHA2-384 and SHA2-512. **Key management technique:** ESV (Cert #98), SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), KAS (Cert #A480) – KAS1-basic with 4096-bit modulus, OneStep KDF with SHA2-512, AES (Cert #C1707) – KWP mode with 256-bit key. **Authentication technique:** Single-Factor Crypto Software.	DRBG Key, DRBG Seed, DRBG V. HA_{PUB}, HA_{PK}, DRBG Key, DRBG Seed, DRBG V. RND, K_{sess}, AEK, AEK-EK, AccessID.	HSM SO, Administrator, AU, Partition SO, Partition CO, Partition LCO, Partition CU.	**E:** DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID. **W:** HA_{PUB}, HA_{PK}, DRBG Key, DRBG Seed, DRBG V. **G:** RND, K_{sess}.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Login as role.	This service is used to login as a given role to a session setup between the client and a target partition. Following successful login, the authentication state for the associated session will be changes to that of the successfully authenticated role. Following login, the authentication state of the session is used to session is used to check and track privileges associated wit	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key, SHA (Cert #C1707) – SHA2-256. **Key management technique:** PBKDF (Cert #A480), KBKDF (Cert #C1707). **Authentication technique:** Memorized Secret, Multi-Factor Crypto Device.	KEK, PEK (if password authentication used), PED Authentication Data, User Password (if challenge-secret enabled), Password, recovered USK, PSK, KCV, GSK, SMK (if configured) (on successful presentation of correct login credentials), AEK, AEK-EK, AccessID.	HSM SO, Administrator, AU, Partition SO, Partition CO, Partition LCO, Partition CU.	**E:** KEK, PEK (if password authentication used), PED Authentication Data, User Password (if challenge-secret enabled), Password, AEK, AEK-EK, AccessID. **W:** recovered USK, PSK, KCV, GSK, SMK (if configured) (on successful presentation of correct login credentials).	IND_1.
Close authenticated sessions.	The service closes authenticated sessions on the request of the user.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	USK, PSK, KCV, GSK, SMK (if configured), AEK, AEK-EK, AccessID, Asymmetric Key Pairs (session keys), Symmetric Keys (session keys).	Any role.	**Z:** USK, PSK, KCV, GSK, SMK (if configured), Asymmetric Key Pairs (session keys), Symmetric Keys (session keys).	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Luna PED Configuration						
Initialize Remote PED Vector (RPV).	This service triggers creation of the module Remote PED Vector. As part of this service, keys used by the PED for remote PED setup are written to an orange PED key connected to Thales Luna PED during initiation of the service.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** ESV (Cert #98), CKG, SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), ECDSA (Cert #C1707) – Key Generation and Signature Generation over curve P521. **Authentication technique:** Single-Factor Crypto Software, Multi-Factor Crypto Device.	GSK, RPV, RPV-C, RPV-K, PED-SKA-C and PED-SKA-K, AEK, AEK-EK, AccessID.	HSM SO.	**E:** GSK, AEK, AEK-EK, AccessID. **G:** PED-SKA-C and PED-SKA-K. **W:** RPV, RPV-C, RPV-K.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Setup Local PED Session.	This service is used to derive a number of shared keys between the module and a Thales Luna PED connected to the modules USB port.	**Algorithms:** N/A. **Key management technique:** ESV (Cert #98), SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), KAS (Cert #A480) – Derivation: ephemeralUnified using curve P521 with full key validation, key-pair generation, KDF: OneStep using SHA2-512, Key Confirmation: HMAC-SHA2-512 with 256-bit key. **Authentication technique:** Single-Factor Crypto Software, Multi-Factor Crypto Device.	PED-EKA-C, HSM-SKA-K_{LOCAL}, CWK_{HSM}, CWK_{PED}, PED Master Shared Secret, AEK, AEK-EK, AccessID.	Any role.	**E:** PED-EKA-C, HSM-SKA-K_{LOCAL}, AEK, AEK-EK, AccessID. **G:** PED Master Shared Secret. **W:** CWK_{HSM}, CWK_{PED}.	IND_2.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Setup Remote PED Session.	This service is used to derive a number of shared keys between the module and a remote Thales Luna PED.	**Algorithms:** N/A. **Key management technique:** ESV (Cert #98), CKG, SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), KAS (Cert #A480) – Derivation: fullUnified using curve P521 with full key validation, key-pair generation, KDF: OneStep using SHA2-512, Key Confirmation: HMAC-SHA2-512 with 256-bit key, ECDSA (Cert #C1707) – signature generation and validation using P521 and SHA2-512, SHA (Cert #C1707) – SHA2-512. **Authentication technique:** N/A.	ECC MIC, ECC-HOC$_{\text{PED}}$, PAC, RPV-C, PED-SKA-C, PED-EKA-C HSM-SKA-K$_{\text{LOCAL}}$, HSM-EKA-C, **G:** PED Master Shared Secret, CWK$_{\text{HSM}}$, CWK$_{\text{PED}}$, DEK$_{\text{HSM}}$, DEK$_{\text{PED}}$, DMK$_{\text{HSM}}$, AEK, AEK-EK, AccessID.	Any role.	**E:** ECC MIC, ECC-HOC$_{\text{PED}}$, PAC, RPV-C, PED-SKA-C, PED-EKA-C HSM-SKA-K$_{\text{LOCAL}}$, HSM-EKA-C, AEK, AEK-EK, AccessID. **G:** HSM-EKA-K, HSM-EKA-C, PED Master Shared Secret, DEK$_{\text{HSM}}$, DEK$_{\text{PED}}$, DMK$_{\text{HSM}}$, CWK$_{\text{HSM}}$, CWK$_{\text{PED}}$. **W:** DEK$_{\text{HSM}}$, DEK$_{\text{PED}}$, DMK$_{\text{HSM}}$, CWK$_{\text{HSM}}$, CWK$_{\text{PED}}$.	IND_2.
Send or receive data over PED tunnel (local PED).	This service is used following setup of an appropriate local PED tunnel in order to transmit encrypted authentication data over the PED tunnel.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** N/A. **Authentication technique:** N/A.	CWK$_{\text{HSM}}$, CWK$_{\text{PED}}$.	Any role.	**E:** CWK$_{\text{HSM}}$, CWK$_{\text{PED}}$.	IND_2.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Send or receive data over PED tunnel (remote PED)	This service is used following setup of an appropriate remote PED tunnel in order to transmit encrypted authentication data over the PED tunnel.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key, CTR mode with 256-bit key, HMAC (Cert #C1707) – HMAC-SHA2-256, SHA (#C1707) – SHA2-256. **Key management technique:** N/A. **Authentication technique:** N/A.	DEK_{HSM}, DEK_{PED}, DMK_{PED} CWK_{HSM}, CWK_{PED}.	Any role.	**E:** DEK_{HSM}, DEK_{PED}, DMK_{PED}, CWK_{HSM}, CWK_{PED}.	IND_2.
Key Management Activities						
Generate local symmetric or asymmetric key-pair.	This service is used to generate symmetric keys or asymmetric key pairs requested by the end-user and stored in the cryptographic module for use with other user consumable cryptographic services or to export to other cryptographic modules or systems.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** ESV (Cert #98), CKG, SHA #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), RSA (Cert #C1707) – Key Generation (all supported methods and curves), ECDSA (Cert #C1707 - Key Generation (all supported methods and curves). **Authentication technique:** N/A.	USK, DRBG Key, DRBG Seed, DRBG V, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID.	HSM SO, Administrator (for admin partition keys). Partition CO, Partition LCO (for user partition).	**E:** USK, DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID. **W:** Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), DRBG Key, DRBG Seed, DRBG V.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Generate domain parameters.[23]	This service is used to generate domain parameters requested by the end-user and stored in the cryptographic module for use with other user consumable cryptographic services or to export to other cryptographic modules or systems.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** ESV (Cert #98), CKG, SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), DSA (Cert #C1707) – Domain parameter generation. **Authentication technique:** N/A.	DRBG Key, DRBG Seed, DRBG V. DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID.	Any role.	**E:** DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID. **W:** DRBG Key, DRBG Seed, DRBG V.	IND_1.
Derive key from existing partition secret or private key object.	This service is used to derive keys based on other key material stored in the module or supplied to it on request of the end-user. Derived keys are stored in the cryptographic module for use with other user consumable cryptographic services or to export to other cryptographic modules or systems.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** ESV (Cert #98), SHA #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), KBKDF (Cert #C1707), KAS-ECC-SSC (Cert #A478), KAS-FFC-SSC (Cert #A478), KDA (Cert #A480), CVL (Cert #A480) – X9.42 KDF. **Authentication technique:** N/A.	Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), USK, DRBG Key, DRBG Seed, DRBG V. Symmetric Keys (general partition or session keys), DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition CO, Partition LCO	**E:** Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), USK, DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID. **W:** Symmetric Keys (general partition or session keys), DRBG Key, DRBG Seed, DRBG V.	IND_1.

[23] Public users cannot generate any objects where either CKA_SENSITIVE or CKA_PRIVATE attributes are true. As such, the service would not affect the security of the module or the security of the information being protected, as sought by 4.1.A.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Import public key, certificate, domain object or data objects.[24]	This service is used to import public key, certificate, domain object or data objects. When importing objects, if the CKA_PRIVATE or CKA_SENSITIVE key attribute it set to true, the object will not be visible to the public user following creation.	**Algorithms: N/A.** **Key management technique:** N/A. **Authentication technique:** N/A.	Asymmetric Key Pairs (general partition or session keys), AEK, AEK-EK, AccessID.	Any role.	**W:** Asymmetric Key Pairs (general partition or session keys). **E:** AEK, AEK-EK, AccessID.	IND_1.
Import secret or private key using key wrapping.	This service is used to import secret or private key from the admin or user partitions using key wrapping. Unauthenticated symmetric encryption is permitted for key unwrapping under Uses allowances in [FIPS 140-3 IG] D.G, Key transport methods.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** KTS-RSA (Cert #A480) – KTS-RSA-OAEP-basic with any supported key size, SHA (Cert #C1707) – any supported hash, AES (Cert #C1707) – all supported key sizes and modes ECB, CBC, CTR, KW, KWP, Triple-DES (Cert #C1707) – all supported key sizes and modes ECB, CBC, CTR. **Authentication technique:** N/A.	Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), USK. Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition CO, Partition LCO.	**E:** Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), USK, AEK, AEK-EK, AccessID. **W:** Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys).	IND_1.

[24] Public users cannot generate/import any objects where either CKA_SENSITIVE or CKA_PRIVATE attributes are true. As such, the service would not affect the security of the module or the security of the information being protected, as sought by 4.1.A.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Export secret or private key using key wrapping.	This service is used to export secret or private key from the admin or user partitions using key wrapping.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** ESV (Cert #98), SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707), KTS-RSA (Cert #A480) – KTS-RSA-OAEP-basic with any supported key size, AES (Cert #C1707) – KW and KWP modes. **Authentication technique:** N/A.	Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys). Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), DRBG Key, DRBG Seed, DRBG V, USK. DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition CO, Partition LCO	**E**: Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), DRBG Key, DRBG Seed, DRBG V, USK, AEK, AEK-EK, AccessID. **R**: Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys). **W:** DRBG Key, DRBG Seed, DRBG V.	IND_1.
Read non-sensitive key attribute where CKA_PRIVATE = false for a given key object.	This service is used to read key attributes to public objects stored by users in the admin or user partition on the cryptographic module.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.
Read non-sensitive key attribute where CKA_PRIVATE = true for a given key object.	This service is used to read key attributes to objects stored by users in the admin or user partition on the cryptographic module and marked as private.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	**E:** AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Insert key from external storage using SKS.	This service is used to import key objects previously extracted from a Thales Luna cryptographic module using either the SIM or SKS feature of the HSM for external storage. SKS inserted keys are encrypted under a key never exposed outside the cryptographic module. Three formats of objects for import are support: > SIM2+ is used with SKS for external storage of keys as the latest format; and > SIM2 and SIM3 are legacy formats supported for import (exclusively) of keys historically extracted from older cryptographic modules.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** SIM2+ Format Objects: AES (Cert #C1707) – GCM mode with 256-bit key. SIM3 Format Objects: AES (Cert #C1707) – CBC mode with 256-bit key, SHA (Cert #C1707) – SHA2-256. SIM2 Format Objects: AES (Cert #C1707) – CTR mode with 256-bit key, SHA (Cert #C1707) – SHA1. **Authentication technique:** N/A.	SMK, USK. Symmetric Keys (general partition or session keys) or Asymmetric Key Pairs (general partition or session keys), AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	**E:** SMK, USK, AEK, AEK-EK, AccessID. **W:** Symmetric Keys (general partition or session keys) or Asymmetric Key Pairs (general partition or session keys)	IND_1.
Extract key to external storage using SKS.	This service is used to export key objects from the module using the SKS feature to extract key objects for external storage. SKS extracted keys are encrypted under a key never exposed outside the cryptographic module. One format for objects extracted from the module is supported: > SIM2+ is used with SKS for external storage of keys.	**Algorithms:** AES (Cert #C1707) – KWP mode with 256-bit key, HMAC (Cert #C1718) – HMAC-SHA2-256, SHA (Cert #C1718) – SHA2-256, SHA2-384. **Key management technique:** AES (Cert #C1707) – GCM mode with 256-bit key. **Authentication technique:** N/A.	SMK, USK. Symmetric Keys (general partition or session keys) or Asymmetric Key Pairs (general partition or session keys), AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	**E:** SMK, USK, AEK, AEK-EK, AccessID. **R:** Symmetric Keys (general partition or session keys) or Asymmetric Key Pairs (general partition or session keys).	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Cryptographic Services						
Re-seed partition DRBG.	This service is used by a user to trigger a manual re-seed operation of a partition DRBG instance.	**Algorithms:** N/A. **Key management technique:** ESV (Cert #98), SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707). **Authentication technique:** N/A.	DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU.	**E/W:** DRBG Key, DRBG Seed, DRBG V. **E:** AEK, AEK-EK, AccessID	IND_1.
Extract entropy from partition DRBG.	This service is used by a user to request and export entropy from a partition DRBG.	**Algorithms:** N/A. **Key management technique:** ESV (Cert #98), CTR_DRBG (Cert #C1707). **Authentication technique:** N/A.	DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU.	**E/W:** DRBG Key, DRBG Seed, DRBG V**.** **E:** AEK, AEK-EK, AccessID	IND_1.
Perform digest operation on user supplied data.	This service is used by a user to request a hash over a block of supplied data.	**Algorithms:** SHA (Cert #C1707) – all hash options supported, SHA3 (Cert #C1707) – all hash options supported. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU.	**E:** AEK, AEK-EK, AccessID.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Perform encrypt operation on user supplied data object.	This service is used by a user to request encryption of a block of user-supplied data using a module stored cryptographic key. Ciphertext resulting from the service is returned the user and not stored.	**Algorithms:** AES (Cert #C1707) – all supported modes and key sizes. **Key management technique:** ESV (Cert #98), SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707) – X9.63 KDF. **Authentication technique:** N/A.	USK (if request requires access to key in non-volatile storage), Symmetric Keys (general partition or session keys), DRBG Key, DRBG Seed, DRBG V. DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	**E:** USK (if request requires access to key in non-volatile storage), Symmetric Keys (general partition or session keys), DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID. **W:** DRBG Key, DRBG Seed, DRBG V.	IND_1.
Perform decrypt operation on user supplied data object.	This service is used by a user to request decryption of a block of user-supplied data using a module stored cryptographic key. Plaintext resulting from the service is returned the user and not stored.	**Algorithms:** AES (Cert #C1707) – all supported modes and key sizes, Triple-DES (Cert #C1707) – all supported modes and key sizes. **Key management technique:** N/A. **Authentication technique:** N/A.	USK (if request requires access to key in non-volatile storage), Symmetric Keys (general partition or session keys), AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU.	**E:** USK (if request requires access to key in non-volatile storage), Symmetric Keys (general partition or session keys).	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Generate signature or MAC over user supplied data.	This service is used by a user to request a signature or MAC over a block of user supplied data (or optionally a user supplied hash for signatures) using a module stored cryptographic key. The resulting signature from the operation is returned the user and not stored.	**Algorithms:** RSA (Cert #C1707), ECDSA (Cert #C1707), DSA (Cert #C1707), HMAC (Cert #C1707), CMAC (Cert #C1707) – AES and Triple-DES, AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** ESV (Cert #98), SHA (Cert #C1707) – SHA2-512, CTR_DRBG (Cert #C1707). **Authentication technique:** N/A.	USK (if request requires access to key in non-volatile storage), Symmetric Keys (general partition or session keys) or Asymmetric Key Pairs (general partition or session keys), DRBG Key, DRBG Seed, DRBG V. Symmetric Keys (general partition or session keys), DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU	**E:** USK (if request requires access to key in non-volatile storage), Symmetric Keys (general partition or session keys) or Asymmetric Key Pairs (general partition or session keys), DRBG Key, DRBG Seed, DRBG V, AEK, AEK-EK, AccessID. **W:** DRBG Key, DRBG Seed, DRBG V.	IND_1.
Validate signature or MAC over user supplied data.	This service is used by a user to request validation of a signature or MAC over a block of user-supplied data using a module stored cryptographic key. The service returns whether the validation was successful.	**Algorithms:** RSA (Cert #C1707), ECDSA (Cert #C1707), DSA (Cert #C1707), HMAC (Cert #C1707), CMAC (Cert #C1707) – AES and Triple-DES, AES (Cert #C1707) – KWP mode with 256-bit key. **Key management technique:** N/A. **Authentication technique:** N/A.	USK (if request requires access to key in non-volatile storage), Symmetric Keys (general partition or session keys) or Asymmetric Key Pairs (general partition or session keys), AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition SO[25], Partition CO, Partition LCO, Partition CU.	**E:** USK (if request requires access to key in non-volatile storage), Symmetric Keys (general partition or session keys) or Asymmetric Key Pairs (general partition or session keys), AEK, AEK-EK, AccessID.	IND_1.

[25] Partition SO is able to validate signatures using public keys where `CKA_PRIVATE` is set to `0`.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Bootloader Services						
Set/read scratchpad flag to signal to main firmware.	This service is used to set flags or read a series of flags in a dedicated FRAM chip on the module. Flags can be used to trigger erasure on boot of option 1: a loaded FM and, option 2: the SMFS, non-volatile storage area available to FM. This feature is used (when enabled) to allow recovery of an HSM that is failing to start due to a malfunctioning FM during startup.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	None.	Any role.	None.	IND_3.
Request complete erase of the HSM main firmware image, loaded FM and key stores (excludes erase of bootloader).	This service is used to recover from corrupt main firmware, or FM and is performed as a factory operation. Following erase, the card needs to repeat manufacturing process including loading factory signed keys before it can be operational again.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	Asymmetric Key Pairs (general partition or session keys).	Any role.	None.	IND_3.
Read Vital Product Data programmed at manufacture.	This service is used to read product data set at manufacture in a EEPROM device such as the HW ID and module serial number.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	None.	Any role.	None.	IND_3.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Request authentication and execution of main firmware.	This service is used to launch the main firmware for the module following successful validation by the bootloader.	**Algorithms:** RSA (Cert #A3164) – RSA PKCS #1 v1.5 signature validation with modulus length 4096, SHA (Cert #C1701) – SHA2-384. **Key management technique:** N/A. **Authentication technique:** N/A.	Root Certificate and Firmware Signing Certificate.	Any role.	**E:** Root Certificate and Firmware Signing Certificate.	IND_3.
Functionality Module Management						
Download FM.	This service is used to download an FM to the module. FM are signed with the signature checked on load against a nominated public key from the admin partition.	**Algorithms:** RSA (Cert #C1707) – PKCS #1-v1.5 signature with all supported key sizes, SHA (Cert #C1707) – SHA2-512. **Key management technique :** N/A. **Authentication technique :** N/A.	Asymmetric Key Pairs (general partition or session keys) – public component of a token stored key, AEK, AEK-EK, AccessID.	HSM SO.	**E:** Asymmetric Key Pairs (general partition or session keys) – public component of a token stored key, AEK, AEK-EK, AccessID.	IND_1.
Activate SMFS.	This service is used to unlock a secure storage capability for data available to FM. Service generates the SMFSK on first activation of the SMFS.	**Algorithms:** AES (Cert #C1707) –KWP mode with 256-bit key. **Key management technique:** ESV (Cert #98), CKG, SHA #C1707) – SHA2-512, CTR_DRBG (Cert #C1707). **Authentication technique:** N/A.	PSK, TVK, SMFSK$_{ENC}$, SMFSK$_{INT}$, AEK, AEK-EK, AccessID.	HSM SO, Administrator.	**E:** PSK, TVK, AEK, AEK-EK, AccessID. **G/W:** SMFS$_{ENC}$, SMFS$_{INT}$.	IND_1.

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Store/retrieve data from SMFS.[26]	This service is used to store and retrieve data in a secure storage capability for data available to FM. Data is encrypted using AES in CTR mode with 256-bit key and stored with a separate MAC to protect its integrity.	**Algorithms:** AES (Cert #C1707) – CTR mode and 256-bit key, HMAC (Cert #C1707) – HMAC-SHA2-256, SHA (Cert #C1707) – SHA2-256. **Key management technique:** N/A. **Authentication technique:** N/A.	$SMFS_{ENC}$, $SMFS_{INT}$, AEK, AEK-EK, AccessID.	Any role (once activated).	**E:** $SMFS_{ENC}$, $SMFS_{INT}$, AEK, AEK-EK, AccessID.	IND_1.
Delete FM.	This service is used to delete an FM.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.
Delete SMFS.	This service is used to delete the contents of the entire SMFS.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.
Get FM status.	This service is used to retrieve status information on a loaded FM.	**Algorithms:** N/A. **Key management technique:** N/A. **Authentication technique:** N/A.	AEK, AEK-EK, AccessID.	Any role.	**E:** AEK, AEK-EK, AccessID.	IND_1.

[26] The FM must first be activated by the HSM SO in order to store/retrieve data from SMFS.

4.4 Non-Approved Services

Non-approved services listed in the table below are not available when the module has been configured to operate in the approved mode (see section 13.2).

As notes on the content of Table 2-6:

> In the 'Indicator Column':

- IND_1 – **Partition Policy (42) Enable CPv1 is set to** `false`, **Partition Policy (43) Enable non-FIPS Algorithms** is set to `false`, **HSM Policy (52), Restrict FM Privilege** is set to `true`, **HSM Policy (56), Allow User Defined ECC Curves** is set to `false`, AND return code is `CKR_OK`;

> **NOTE** While default setting for **HSM Policy (52), Restrict FM Privilege** is `false`, this setting is only relevant to a FIPS approved configuration following load of the 'Enable Functionality Module CUF'. This unlocks the ability to enable the related '**HSM Policy (50) Allow Functionality Modules**'. Prior to HSM Policy (50) being set to `true`, the setting of HSM Policy (52) Restrict FM Privilege is redundant to the modules operation as FM can't be loaded and the policy only relates to commands received by the module from within code executing within a sandboxed functionality module.

Table 4-5: Non-Approved Services

Service	Description	Non-Approved Algorithms Accessed	Roles	Indicator
Cryptographic Services				
Perform digest operation on user supplied data	This service is used by a user to request a hash over a block of supplied data. This service is not possible for the Backup configuration	HAS-160, KECCAK, MD2, MD5, RIPEMD-160, SM3.	HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU	IND_1
Perform encrypt operation on user supplied data object	This service is used by a user to request encryption of a block of user-supplied data using a module stored cryptographic key. Ciphertext resulting from the service is returned the user and not stored. This service is not possible for the Backup configuration.	ARIA, CAST3, CAST5, DES, RC2, RC4, RC5, RSA[27], RSA X.509, SEED, SM4, Triple-DES, XOR.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU	IND_1
Perform decrypt operation on user supplied data object	This service is used by a user to request decryption of a block of user-supplied data using a module stored cryptographic key. Plaintext resulting from the service is returned the user and not stored. This service is not possible for the Backup configuration.	ARIA, CAST3, CAST5, DES, RC2, RC4, RC5, RSA[28], RSA X.509, SEED, SM4, Triple-DES[29], XOR.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU	IND_1
Generate signature or MAC over user supplied data	This service is used by a user to request a signature or MAC over a block of user supplied data (or optionally a user supplied hash for signatures) using a module stored cryptographic key. The resulting signature from the operation is returned the user and not stored.	**Symmetric Algorithms:** ARIA-CMAC, SEED-CMAC, Triple-DES-CMAC, HMAC[30], HAS160-MAC, MD5-HMAC, SM3-HMAC, RIPEMD160-HMAC, AES-MAC, ARIA-MAC, CAST3-MAC, CAST5-MAC, DES-MAC, RC2-MAC, RC5-MAC, SEED-MAC, SSL3-MD5-MAC, SSL3-SHA1-MAC, Triple-DES-MAC, Triple-DES-x9.19-MAC, TUAK, MILENAGE, COMP128.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU	IND_1

[27] RSA is non-compliant when using PKCS#1, v1.5 padding for encryption or decryption.
[28] RSA is non-compliant with less than 112 bits of encryption strength.
[29] Triple-DES is non-compliant with less than 112 bits of encryption strength.
[30] HMAC is non-compliant with less than 112-bits of encryption strength.

Table 4-5: Non-Approved Services

Service	Description	Non-Approved Algorithms Accessed	Roles	Indicator
	This service is not possible for the Backup configuration.	**Asymmetric Algorithms**: DSA[31], ECDSA[32], EdDSA, EdDSA PH, KCDSA, RSA[33], SM2, SM3.		
Validate signature or MAC over user supplied data	This service is used by a user to request validation of a signature or MAC over a block of user-supplied data using a module stored cryptographic key. The service returns whether the validation was successful. This service is not possible for the Backup configuration.	**Symmetric Algorithms:** ARIA-CMAC, SEED-CMAC, Triple-DES-CMAC[34], HMAC[35], HAS160-MAC, MD5-HMAC, SM3-HMAC, RIPEMD160-HMAC, AES-MAC, ARIA-MAC, CAST3-MAC, CAST5-MAC, DES-MAC, RC2-MAC, RC5-MAC, SEED-MAC, SSL3-MD5-MAC, SSL3-SHA1-MAC, Triple-DES-MAC, Triple-DES-x9.19-MAC, TUAK, MILENAGE, COMP128. **Asymmetric Algorithms**: DSA[36], ECDSA[37], EdDSA, EdDSA PH, KCDSA, RSA[38], SM2, SM3.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU	IND_1
HSM Management				
Clone partition objects between partitions[39]	This service supports use of CPV1 for key object import and export exclusively.	AES, RSA.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU	IND_1
Key Management Activities				
Derive key from existing partition secret or private key object	This service is used to derive keys based on other key material stored in the module or supplied to it on request of the end-user. Derived keys are stored in the cryptographic module for use with other user consumable cryptographic services or to export	AES[40], ARIA, BIP32, DES, MD5, SHA, SSL PRE-MASTER, SSL3-MASTER, SM3, Triple-DES, XOR.	HSM SO, Administrator, Partition CO, Partition LCO	IND_1

[31] DSA is non-compliant with less than 112 bits of encryption strength.

[32] ECDSA is non-compliant with less than 112 bits of encryption strength.

[33] RSA is non-compliant with less than 112 bits of encryption strength.

[34] Triple-DES-CMAC is non-compliant with less than 112-bits of encryption strength.

[35] HMAC is non-compliant with less than 112-bits of encryption strength.

[36] DSA is non-compliant with less than 112 bits of encryption strength.

[37] ECDSA is non-compliant with less than 112 bits of encryption strength.

[38] RSA is non-compliant with less than 112 bits of encryption strength.

[39] this service uses both [SP800-56Br2] non-compliant RSA encryption for encryption of a nonce followed by AES encryption of key objects in a [SP800-38F] non-compliant manner when Cloning Protocol Version 1 is used for key export. Later versions of this protocol as separately mapped to this service use approved cryptography.

[40] AES is non-approved for key derivation when use to derive keys using methods other than as permitted by NIST standard such as [SP800-56Cr2] and [SP800-108r1] in particular, use of AES in ECB or CBC mode directly to derive keys.

Table 4-5: Non-Approved Services

Service	Description	Non-Approved Algorithms Accessed	Roles	Indicator
	to other cryptographic modules or systems. This service is not possible for the Backup configuration.			
Generate local symmetric or asymmetric key-pair	This service is used to generate symmetric keys or asymmetric key pairs requested by the end-user and stored in the cryptographic module for use with other user consumable cryptographic services or to export to other cryptographic modules or systems. This service is not possible for the Backup configuration	Diffie-Hellman[41], ECC[42], KCDSA, RSA[43], SM2.	HSM SO, Administrator, Partition CO, Partition LCO	IND_1
Import secret or private key using key wrapping	This service is used to import secret or private key from the admin or user partitions using key wrapping Unauthenticated symmetric encryption is permitted for key unwrapping under Uses allowances in [FIPS 140-3 IG] D.G, Key transport methods. This service is not possible for the Backup configuration.	ARIA, CAST3, CAST5, DES, RC2, RSA[44], SEED, SM4.	HSM SO, Administrator, Partition CO, Partition LCO	IND_1
Export secret or private key using key wrapping	This service is used to export secret or private key from the admin or user partitions using key wrapping. This service is not possible for the Backup configuration.	AES[45], ARIA, CAST3, CAST5, DES, RC2, RSA, SEED, SM4, TDES.	HSM SO, Administrator, Partition CO, Partition LCO	IND_1

[41] Diffie-Hellman key generation mechanisms are non-compliant with less than 112-bits of encryption strength.

[42] ECC key generation mechanisms are non-compliant with less than 112-bits of encryption strength.

[43] RSA key generation mechanisms are non-compliant with less than 112-bits of encryption strength.

[44] RSA is non-approved for key transport when used with an encryption strength of 112-bits or when using PKCS #1, v1.5 padding for encryption.

[45] AES is non-approved for key transport when used to encrypt keys using methods other than as permitted by NIST standards such as [SP800-38F]. In particular, use of un-authenticated modes of AES for encryption without a separate authentication tag (e.g. signature or MAC) is non-approved.

Table 4-5: Non-Approved Services

Service	Description	Non-Approved Algorithms Accessed	Roles	Indicator
Generate domain parameters.[46]	This service is used to generate domain parameters requested by the end-user and stored in the cryptographic module for use with other user consumable cryptographic services or to export to other cryptographic modules or systems.	X9.42 Domain Parameter Generation	Any role	IND_1

[46] Public users cannot generate any objects where either `CKA_SENSITIVE` or `CKA_PRIVATE` attributes are true. As such, the service would not affect the security of the module or the security of the information being protected, as sought by 4.1.A.

5 Software/Firmware Security

5.1 Firmware Integrity

The Thales Luna K7 Cryptographic Module's firmware integrity is checked on startup as described in section 10.1. The bootloader runs the self-test functions to check the firmware integrity as well as the cryptographic algorithms used to check the bootloader and main firmware image authenticity. Any failures during these tests will result in a module halt in which an error message is output, the module halts all functions and data output is inhibited. The main firmware image is used to check the integrity of any loaded FM.

The bootloader and firmware are stored as signed binaries using RSA PKCS #1-v1.5 with a 4096-bit module and SHA2-384.

The operator can trigger an on-demand check of the module firmware using the `CA_SelfTest` Cryptoki API command. An example of the `CA_SelfTest` Cryptoki API command in use can be found in section 13.3.

Periodic Self-Tests (PST) are performed every 24 hours and run the firmware integrity tests and a subset of the KAT tests. Failure of either of these self-tests during PST will trigger a module halt. Recovery from this state will require the module to be restarted and for the detected fault to have cleared otherwise the module will re-halt during POST following restart. See Section 10 for additional information about the PST.

5.2 FM Integrity

All FM images downloaded into the module must have an assigned signature from the FM developer. FMs are only executed inside the HSM after this signature has been validated.

The FM code authenticity and integrity is protected using RSA #1 v1.5 signature with SHA2-512 digest and a minimum of 2048-bit modulus.

5.3 Firmware Load

When new main firmware is to be loaded using the `hsm updatefw` LunaCM command, a separate mechanism is used to authenticate the firmware than the pre-operational firmware self-test. An example of the `hsm updatefw` LunaCM command can be found in section 11.11.

Once initiated the firmware load sequence uses a set series of ICD commands and all others are prohibited until the firmware update is completed.

Updating the main firmware is a two stage process. The first stage is to download the main firmware to the module. The second stage involves subsequently re-authenticating and loading the main firmware following a module restart, this occurs based on the bootloader during power-on ahead of the communications module being started.

The firmware load test can be found in Table 10-2.

5.4 Firmware Components

The following are the firmware components included on the module:

> `hsm` - this component is compiled as a 32bit LSB executable for PowerPC. This is identified throughout this document as the 'main firmware'.

> `bootloader` - the bootloader is an Executable and Linkable Format (ELF) executable. This is identified throughout this document as the 'bootloader'.

No source code, object code or just-in-time compiled code are included in the module.

This scope of this certification does not currently include any certified FM where these are expected to have independent FIPS 140-3 certificates. As such, no components covering FM are listed in this section.

6 Operational Environment

The module supports a **limited operating environment** as defined in [ISO/IEC 19790:2012].

The only changes to the environment permitted in the approved mode of operation is through the loading of FM where the loaded FM has been [FIPS 140-3] certified to run on Thales Luna K7 Cryptographic Module.

7 Physical Security

7.1 Mechanism Summary

7.1.1 Module Construction

The module is of physical embodiment **multi-chip embedded** and supports **physical security level 3.** The module is enclosed in a strong metal enclosure that provides tamper-evidence. Any tampering that might compromise a module's security is detectable by visual inspection of the physical integrity of a module. The HSM SO should perform a visual inspection of the module at regular intervals.

Within the metal enclosure, a hard opaque epoxy covers the circuitry of the cryptographic module. Attempts to remove this epoxy will cause sufficient damage to the cryptographic module so that it is rendered inoperable.

The module's enclosure is opaque to resist visual inspection of the device design, physical probing of the device and attempts to access sensitive data on individual components of the device.

7.1.2 External Event trigger

The module supports a physical interface for the input of an external event signal. The external event signal jumper is monitored in both the powered-on state and the powered-off state.

In the event of an external event signal, the module will erase the Token Module Variable Key, reset itself, clear all working memory and log the event. The module can be reset and placed back into operation when the external event signal is removed.

7.1.3 PCIe Card Removal

The module detects removal from the PCI-E slot in both the powered-on state and the powered-off state. If the card is removed from the PCI-E slot, the Token Module Variable Key (TVK) is erased and the event is logged.

7.1.4 Environment Failure Protection

The module supports an EFP mechanism that will trigger module shutdown if low or high temperature extremes and out-of-range voltage conditions are detected whilst the module is active. This is covered in more detail in section 7.3.

7.2 Module Inspection

The following routine inspections are recommended.

Table 7-1: Physical Security Inspection Guidelines

Physical Security Mechanism	Recommended Frequency of Inspection/Test	Inspection/Test Guidance Details
Physical inspection of HSM surfaces for signs of tamper.	On receipt of HSM following transport; At any point following any un-authorized access to the environment hosting the HSM; and Following any extended periods of unattended storage for the module.	<see below>.

Following manufacture, both the front and rear covers of the Thales Luna K7 Cryptographic Module are permanently adhered to the PCB assembly using epoxy resin and with the lid assemblies having feet set into the epoxy.

Any attempts to remove the covers will result in significant physical damage to the card rendering it un-usable.

Example (but not exhaustive) pictures of potential attempts to tamper a card are shown in the figure below:

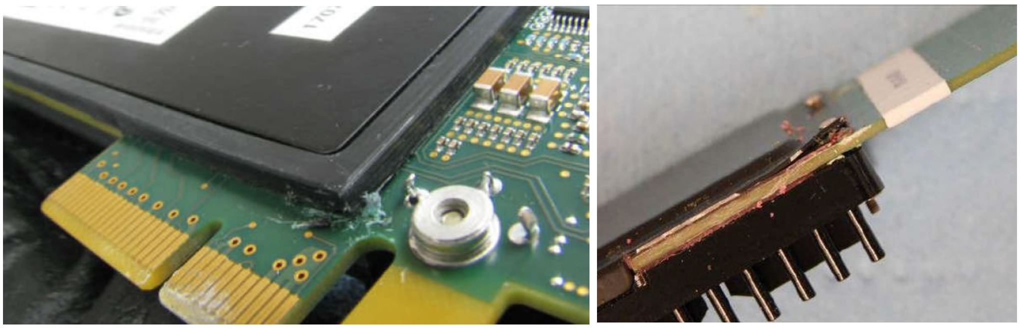

Figure 7-1: Example indicators of a tamper event during shipping (prizing at module corners).

In the event of any observed damage, photograph the card and contact Thales to confirm whether observed anomalies are to be expected or are confirmed signs of potential tampering.

7.3 Environment Failure Protection

The module's hardware is designed to sense and respond to out-of-range temperature conditions as well as out-of-range voltage conditions. The temperature and voltage conditions are only monitored in the powered-on state.

In the event that the module senses an out-of-range temperature or over voltage, the module will erase the TVK, reset itself, clear all working memory and log the event.

The module can be reset and placed back into operation when in-bound operating conditions have been restored

The module monitors three voltage rails: 5V, 3.3V and 1.8V each of which can independently trigger an EFP event. The following table covers the limits enforced by the module:

Table 7-2: EFP/EFT

	Temperature or voltage measurement	Specify EFP or EFT	Specify if this condition results in a shutdown or zeroisation
Low Temperature	-2ºC	EFP	shutdown.
High Temperature	+80ºC	EFP	shutdown.
Low Voltage	5V net – 4.76V, 3.3V net – 3.07V, 1.8V net – 1.62V.	EFP	shutdown.
High Voltage	5V net – 5.26V, 3.3V net – 3.61V, 1.8V net –2V.	EFP	shutdown.

7.4 Module Coatings

The module PCB is potted using an epoxy-based compound inside the area identified as cryptographic boundary in Figure 2-1 and Figure 2-2. The potting compound is applied directly to the PCB inside a fence. Heatsinks are anchored to the potted area during the curing process.

The following table lists the temperature range tested during the assessment of the module.

Table 7-3: Hardness testing temperature ranges

	Hardness tested temperature measurement
Low Temperature	-20ºC
High Temperature	+80ºC

> **NOTE** Hardness temperature covered the temperature range permitted for storage of the module. The modules operational temperature range sits within these values.
>
> Further details on the operating ranges for the module for both input voltage and temperature can be found in section 13.4.

8 Non-invasive security

N/A: [ISO/IEC 19790:2012] Section 6.8, Non-invasive security is non-applicable as there are currently no requirement in [SP800-140F].

9 SSP Management

9.1 Sensitive Security Parameter

The following table lists Sensitive Security Parameters (SSP) used to perform approved security function supported by the cryptographic module.

The following notes should be observed when reading the table:

> When reading the ’zeroization’ column, the following mapping for listed overwrite methods should be used:

- KDM1: Overwrite memory containing the key material;
- KDM2: RAM reset;
- KDM3: Erasure of entire memory sector(s);
- KDM4: Erasure of the encrypting keys; and
- KDM5: Erasure of the HSE-BBRAM in response to a tamper/decommission event.

NOTE A HSM wipe-out (KDM3) zeroizes all keys and CSPs on the module. This method applies to every row in Table 9-1 and is explicitly called out in the table only if the SSP is not covered by any other destruction method.

> When reading the ‘strength’ column, the listed security strength is calculated using methods in [FIPS 140-3 IG] D.B, ‘Strength of SSP Establishment Methods’.

> When reading the ‘Security Function and Cert Number’ column, this is the security function that will consume the SSP.

> Details on schemes covered under the ‘Key Import/Export methods’ column are expanded in section 9.3.

Table 9-1: Summary of SSPs

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
Key Encryption Key (KEK) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	[SP800-90Ar1] CTR_DRBG with AES-256.	Not Input or Output.	N/A.	HSE-BBRAM in plaintext	Zeroized in response to decommission – KDM5	When administrators enable Decommission, the KEK encrypts all sensitive values and is zeroized in response to a decommission signal. **This key is a CSP.**
Root Certificate public key certificate 4096-bit.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	N/A – generated outside the module.	Loaded at manufacture as part of the bootloader image. Certificate output in plaintext.	N/A.	Flash memory in plaintext	Full HSM Wipe - KDM3	The X.509 public key certificate corresponding to the Root Key. It is self-signed with its private key controlled by Thales. Used in verifying Manufacturing Integrity Certificate (MIC), firmware and capability updates. **This key is a PSP.**
Firmware Signing Certificate 4096-bit public key certificate.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	N/A – generated outside the module.	Input with Firmware Update Image, which is considered plaintext.	N/A.	Flash memory in plaintext	Full HSM Wipe - KDM3	The X.509 public subordinate certificate signed by “Root Private” signing key used to certify HSM firmware updates. **This key is a PSP.**
Capability Signing Certificate 4096-bit public key certificate.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	N/A – generated outside the module.	Input with Capability Update File.	N/A.	Flash memory in plaintext	Full HSM Wipe - KDM3	The X.509 public subordinate certificate signed by “Root Private” signing key used to certify HSM capability updates. **This key is a PSP.**
Manufacturer’s Integrity Certificate (MIC) 4096-bit public key certificate.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	N/A – generated outside the module.	Certificate Output in Plaintext.	N/A.	Flash memory in plaintext	Full HSM Wipe - KDM3	The X.509 public key certificate corresponding to the Manufacturing Integrity Key (MIK) controlled by Thales. It is signed by the Root Key. Used in verifying all key material certified by Hardware Origin Certificates (HOCs). **This key is a PSP.**
ECC Manufacturing Integrity Certificate (ECC MIC) ECC public certificate for public key on curve P-384.	192-bit.	ECDSA (Cert #C1707 and #C1718).	N/A – generated outside the module.	Certificate Output in Plaintext.	N/A.	Flash memory plaintext	Full HSM Wipe - KDM3	The X.509 public key certificate corresponding to the ECC Manufacturing Integrity Key (ECC MIK). It is self-signed. **This key is a PSP.**
Hardware Origin Key (HOK) 4096-bit private key.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Not Input or Output.	N/A.	Flash memory encrypted with GSK	Full HSM Wipe - KDM3 Erased when FM policy is enabled – KDM3	A 4096-bit RSA private key used to sign certificates for other device key pairs, such as the TWC4 used with CPV3. It is generated at the time the device is manufactured. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
Hardware Origin Certificate (HOC) 4096-bit public key certificate.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	Loaded at manufacturing	Certificate Output in Plaintext.	N/A.	Flash memory in plaintext	Full HSM Wipe - KDM3 Erased when FM policy is enabled – KDM3	The X.509 public key certificate corresponding to the HOK. It is signed by the Manufacturer's Integrity Key (MIK) at the time the device is manufactured. Used in verifying all key material signed by the HOK. **This key is a PSP.**
FM Hardware Origin Key (FM HOK) 4096-bit private key.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Not Input or Output.	N/A.	Flash memory encrypted with GSK	Full HSM Wipe - KDM3	A 4096-bit RSA private key used to sign certificates for other device key pairs, such as the TWC4. It is generated at the time the device is manufactured. **This key is a CSP.**
FM Hardware Origin Certificate (FM HOC) 4096-bit public key certificate.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	Loaded at manufacturing	Certificate Output in Plaintext.	N/A.	Flash memory in plaintext	Full HSM Wipe - KDM3	The X.509 public key certificate corresponding to the FM HOK. It is signed by the Manufacturer's Integrity Key (MIK) at the time the device is manufactured. Used in verifying all key material signed by the FM HOK. **This key is a PSP.**
ECC Hardware Origin Key (ECC HOK) ECC private key on curve P-384.	192-bit.	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output.	N/A.	Flash memory encrypted with GSK	Full HSM Wipe - KDM3 Erased when FM policy is enabled – KDM3	ECC P-384 private key used to sign other device keys and used for a specific PKI implementation requiring assurance that a key or a specific action originated within the hardware crypto module. **This key is a CSP.**
ECC Hardware Origin Certificate (ECC HOC) ECC public certificate for public key on curve P-384.	192-bit.	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Certificate Output in Plaintext.	N/A.	Flash memory plaintext	Full HSM Wipe - KDM3 Erased when FM policy is enabled – KDM3	The X.509 public key certificate corresponding to the ECC HOK. It is signed by the ECC Manufacturing Integrity Key (ECC MIK). It is used for a specific PKI implementation requiring assurance that a key or a specific action originated within the hardware crypto module. **This key is a PSP.**
FM ECC Hardware Origin Key (FM ECC HOK) ECC private key on curve P-384.	192-bit.	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output.	N/A.	Flash memory encrypted with GSK	Full HSM Wipe - KDM3	ECC P-384 private key used to sign other device keys and used for a specific PKI implementation requiring assurance that a key or a specific action originated within the hardware crypto module. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
FM ECC Hardware Origin Certificate (FM ECC HOC) ECC public certificate for public key on curve P-384.	192-bit.	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Certificate Output in Plaintext.	N/A.	Flash memory plaintext	Full HSM Wipe - KDM3	The X.509 public key certificate corresponding to the FM ECC HOK. It is signed by the ECC Manufacturing Integrity Key (ECC MIK). It is used for a specific PKI implementation requiring assurance that a key or a specific action originated within the hardware crypto module. **This key is a PSP.**
Token or Module Unwrapping Key (TUK3) 2048-bit private key.	112-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Not Input or Output.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM1 Erased on user zeroize request or destructive policy change.	A 2048-bit RSA private key used with the Cloning Protocol Version 1 supported for key import only. It is following initial request for the key. **This key is a CSP.**
Token or Module Wrapping Certificate (TWC3) 2048-bit public key certificate.	112-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Certificate Output in Plaintext.	N/A.	Working SDRAM in plaintext	Power Cycle – KDM1. Erased on user zeroize request or destructive policy change	The X.509 public key certificate corresponding to the TUK4. It is signed by the HOK. Used in exchange of nonce (KEVs and KEVt) as part of the handshake during the cloning protocol version 1 supported for key import only. **This key is a PSP.**
Token or Module Unwrapping Key (TUK4) 4096 bit private key.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Not Input or Output.	N/A.	Working SDRAM in plaintext	Power Cycle – KDM1. Erased on user zeroize request or destructive policy change.	A 4096 bit RSA private key used in the key cloning protocol. It is generated each time the module initializes from power up or reset. **This key is a CSP.**
Token or Module Wrapping Certificate (TWC4) 4096 public key certificate.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Certificate Output in Plaintext.	N/A.	Working SDRAM in plaintext	Power Cycle – KDM1. Erased on user zeroize request or destructive policy change.	The X.509 public key certificate corresponding to the TUK4. It is signed by the HOK. Used in exchange of nonce (KEVs and KEVt) as part of the handshake during the cloning protocol. **This key is a PSP.**
Cloning Key Encryption Vector – source (KEV^s) 384 bit nonce.	256-bit.	KDA (Cert #A480).	[SP800-90Ar1] CTR_DRBG with AES-256.	Exchanged during CPV3 protocol (see section 9.3 for further details).	N/A.	Working SDRAM in plaintext.	Zeroized following use – KDM1.	384-bit nonce used with the cloning protocol and generated on the source HSM. **This key is a CSP.**
Cloning Transfer Key 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	OneStep KDF from [SP800-56Cr2].	Not Input or Output.	Established using CPV3 as covered in section 9.3.	Working SDRAM in plaintext.	Zeroized following use – KDM1.	256-bit AES key derived during the cloning protocol and used to transfer key objects between source and target partitions using the cloning protocol. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
CPV4 Key Agreement Source Private Key ECC private key on curve P-521 or BrainpoolP512r1.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output. Single use ephemeral key.	N/A.	Working SDRAM in plaintext.	Zeroized following use – KDM1.	Used to establish the ECDH key agreement between HSMs. **This key is a CSP.**
CPV4 Key Agreement Source Public Key ECC public key on curve P-521 or BrainpoolP512r1.	256-bit	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Exported to a peer HSM as part of the CPV4 protocol. Single use ephemeral key.	N/A.	Working SDRAM in plaintext.	Zeroized following use – KDM1.	Used to establish the ECDH key agreement between HSMs. Signed by the CPV4 Messaging Private Key of the source HSM. **This key is a PSP.**
CPV4 Key Agreement Destination Public Key ECC public key on curve P-521 or BrainpoolP512r1.	256-bit	KAS-ECC (Cert #A480).	N/A – imported from peer HSM.	Exported to a peer HSM as part of the CPV4 protocol. Single use ephemeral key.	N/A.	Working SDRAM in plaintext.	Zeroized following use – KDM1.	Used to establish the ECDH key agreement between HSMs. Signed by the CPV4 Messaging Private Key of the destination HSM. **This key is a PSP.**
CPV4 Key Agreement Shared Secret 256-bit AES key.	256-bit	KDA (Cert #A480).	N/A.	Not Input/output. Single use ephemeral key.	Derived using ECDH OneStep Key Derivation (section 5.8.2.1 from [SP800-56Ar3]) with C(2e, 0s, ECC CDH) with P-521 or BrainpoolP512r1.	Working SDRAM in plaintext	Zeroized following use – KDM1.	Shared secret output from ECDH during CPV4 negotiation. Shared secret is used to derive the CPV4 Key Transport Key. **This key is a CSP.**
CPV4 Key Transport Encryption Key 256-bit AES key.	256-bit	AES (Cert #C1707).	Derived using OneStep KDF from [SP800-56Cr2] and SHA2-512 or SHA3-512 as the PRF. Inputs to the KDF include the CPV4 Key Agreement Shared Secret alongside the Key Cloning Domain Vector (KCV).	Not Input/output. Single use ephemeral key.	N/A.	Working SDRAM in plaintext	Zeroized following use – KDM1.	This key wraps the CPV4 Session Salt sent to the destination HSM and used as an input to the KDF used to derive the CPV4 Per-Blob Encryption Key and CPV4 Per-Blob Mac Keys (when required). **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
CPV4 Key Transport MAC Key 256-bit key.	256-bit	HMAC (Cert #C1707).	Derived using OneStep KDF from [SP800-56Cr2] and SHA2-512 or SHA3-512 as the PRF. Inputs to the KDF include the CPV4 Key Agreement Shared Secret alongside the Key Cloning Domain Vector (KCV).	Not Input/output. Single use ephemeral key.	N/A.	Working SDRAM in plaintext	Zeroized following use – KDM1.	This key is used to MAC the encrypted object containing the CPV4 Session Salt sent to the destination HSM and where the encryption algorithm option used is used as an input to the KDF used to derive the CPV4 per-Blob Encryption Key and CPV4 per-Blob Mac Keys. **This key is a CSP.**
CPV4 Session Key 256-bit key.	256-bit	KDA (Cert #A480).	Derived using OneStep KDF from [SP800-56Cr2] and SHA2-512 or SHA3-512 as the PRF. Inputs to the KDF include the CPV4 Key Agreement Shared Secret alongside the Key Cloning Domain Vector (KCV).	Not Input/output.	N/A.	Working SDRAM in plaintext.	Power Cycle – KDM1. Erased on user zeroize request or destructive policy change.	Session key used as part of the CPV4 protocol as and input to the KDF used to derive the CPV4 Per-Blob Encryption Key and CPV4 Per-Blob MAC Key used to encrypt messages in CPV4. The key is used for the session lifetime (60 minutes) for CPV4 ahead of being re-negotiated and replaced. **This key is a CSP.**
CPV4 Session Salt 256-bit key.	256-bit	KDA (Cert #A480).	[SP800-90Ar1] CTR_DRBG with AES-256.	Exported as part of the CPV4 protocol encrypted under the CPV4 Transport Key using either AES-256 in GCM and with a 128-bit random IV and 128-bit TAG or with AES in CTR mode and using either HMAC-SHA2-512 or HMAC-SHA3-512.	N/A.	Working SDRAM in plaintext.	Power Cycle – KDM1. Erased on user zeroize request or destructive policy change.	Salt value used as input to the KDF used to generate the CPV4 Per-Blob Encryption Key and CPV4 Per-Blob MAC Key. **This key is a CSP.**
CPV4 Per-Blob Salt 256-bit key.	256-bit	KDA (Cert #A480).	[SP800-90Ar1] CTR_DRBG with AES-256.	Output in plaintext as part of CPV4 protocol.	N/A.	Working SDRAM in plaintext.	Power Cycle – KDM1. Erased on user zeroize request or destructive policy change.	Random 256-bit salt unique per object encryption operation and used as input to (alongside the CPV4 Session Salt and CPV4 Session K7) to the KDF used to derive the CPV4 Per Blob Encryption Key and CPV4 Per Blob MAC Key. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
CPV4 Per-Blob Encryption Key 256-bit AES key.	256-bit	AES (Cert #C1707).	Derived using OneStep KDF from [SP800-56Cr2] and either SHA2-512 or SHA3-512 as the PRF. Inputs to the KDF include the CPV4 Session key, CPV4 Session Salt and CPV4 Per-Blob Salt.	Not Input/output. Single use ephemeral key.	N/A.	Working SDRAM in plaintext.	Zeroized following use – KDM1.	Key derived from the CPV4 session key, CPV4 Session Salt and CPV4 Per-Blob Salt and used to encrypt individual keys in transport. The derived key is unique per key transferred. **This key is a CSP.**
CPV4 Per-Blob MAC Key 256-bit HMAC key.	256-bit	HMAC (Cert #C1707).	Derived using OneStep KDF from [SP800-56Cr2] and either SHA2-512 or SHA3-512 as the PRF. Inputs to the KDF include the CPV4 Session key, CPV4 Session Salt and CPV4 Per-Blob Salt.	Not Input/output. Single use ephemeral key.	N/A.	Working SDRAM in plaintext.	Zeroized following use – KDM1.	Key derived from the CPV4 session key, CPV4 Session Salt and CPV4 Per-Blob Salt and used to generate a MAC over individual encrypted keys in transport when AES in CTR mode is selected as the cipher. The derived key is unique per key transferred. **This key is a CSP.**
CPV4 Messaging Private Key ECC private key on either curve P-521 or BrainpoolP512r1.	256-bit	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Not Input/output.	N/A.	Working SDRAM in plaintext	Power Cycle – KDM1. Erased on user zeroize request or destructive policy change.	This signs CPV4 messages and the ephemeral keys. It is signed by either the HOK or FM HOK. **This key is a CSP.**
CPV4 Messaging Public Certificate ECC public key certificate on either curve P-521 or BrainpoolP512r1.	256-bit	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Output as part of the CPV4 protocol to peer HSM.	N/A.	Working SDRAM in plaintext	Power Cycle – KDM1. Erased on user zeroize request or destructive policy change.	Certificate included with signed messages and ephemeral keys when exchanged between HSM during the CPV4 protocol. **This key is a PSP.**
Device Authentication Key (CITS-DAK) 4096-bit private key.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Not Input or Output.	N/A.	Working SDRAM in plaintext	Power Cycle – KDM1. Erased on user zeroize request or destructive policy change.	4096-bit RSA private key used for a specific PKI implementation requiring assurance that a key or a specific action originated within the hardware crypto module. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
Device Authentication Key (CITS-DAC) 4096-bit public key certificate.	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Certificate Output in Plaintext.	N/A.	Working SDRAM in plaintext	Full HSM Wipe - KDM3	The X.509 public key certificate corresponding to the CITS-DAK. Signed by the HOK. Used for a specific PKI implementation requiring assurance that a key or a specific action originated within the hardware crypto module. **This key is a PSP.**
ECC Device Authentication Key (ECC DAK) ECC private key on curve P-384.	192-bit.	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output.	N/A.	Flash memory encrypted with GSK	Full HSM Wipe - KDM3	ECC P-384 private key. **This key is a CSP.**
ECC Device Authentication Certificate (ECC DAC) ECC public certificate for public key on curve P-384.	192-bit.	ECDSA (Cert #C1707 and #C1718).	N/A – generated outside the module.	Certificate Output in Plaintext.	N/A.	Flash memory plaintext	Full HSM Wipe - KDM3	The X.509 public key certificate corresponding to the ECC DAK. It is signed by the ECC HOK. **This key is a PSP.**
Token or Module Variable Key (TVK) 256-bit AES key.	256-bit.	AES (Cert #C1707).	[SP800-90Ar1] CTR_DRBG with AES-256.	Not Input or Output.	N/A.	HSE-BBRAM in plaintext	Zeroized in response to physical security measures - KDM5	It is used to encrypt authentication data stored for auto-activation purposes. **This key is a CSP.**
Secure Transport Mode (STM) Nonce 992-bits.	256-bit.	SHA (Cert #C1707).	[SP800-90Ar1] CTR_DRBG with AES-256.	Not Input or Output.	N/A.	HSE-BBRAM in plaintext	Zeroized in response to physical security measures - KDM5	Random value used to create module fingerprint that is used to verify the module's integrity as part of the Secure Transport Mode feature. **This key is a PSP.**
DRBG Key 256-bit AES key.	256-bit.	CTR_DRBG (Cert #C1707).	Internal state generated using CTR_DRBG from [SP800-90Ar1].	Not Input or Output.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM2	32 bytes AES key stored in the RAM. Used in an implementation of the NIST SP 800-90Ar1 CTR (AES) DRBG. **This key is a CSP.**
DRBG Seed 384 bits.	384-bit.	CTR_DRBG (Cert #C1707).	Full-entropy conditioned output from ESV (Cert #98) approved platform noise source.	Not Input or Output.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM2	Random seed data drawn from the Hardware RBG and used to seed an implementation of the NIST SP 800-90Ar1 CTR (AES) DRBG. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
DRBG V 128 bits.	128-bit.	CTR_DRBG (Cert #C1707).	Internal state generated using CTR_DRBG from [SP800-90Ar1].	Not Input or Output.	N/A.	Working SDRAM in plaintext.	Power Cycle - KDM2	Part of the secret state of the approved DRBG. The value is generated using the methods described in [SP800-90Ar1]. **This key is a CSP.**
Global Storage Key (GSK) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	[SP800-90Ar1] CTR_DRBG with AES-256.	Not Input or Output.	N/A.	Flash memory encrypted with PSK.	Full HSM Wipe - KDM3	256-bit AES key that is the same for all users on a specific Luna cryptographic module. It is used to encrypt permanent parameters within the non-volatile memory area reserved for use by the module. **This key is a CSP.**
Role Domain Key (RDK) 256-bit key.	256-bit. Or 32 to 2040-bit.	KDA (Cert #A480).	[SP800-90Ar1] CTR_DRBG with AES-256 for PED configuration. N/A for Password configuration.	Input / Output via direct connection to PED.	N/A.	Flash Memory encrypted with USK.	Factory Reset - KDM1	For PED configurations, this is a 256-bit value, the first 32-bytes of which are used as an AES KW 256-bit key that is used to wrap/unwrap the SALK when it is exported / imported from / to the module. It is either generated by the module or imprinted onto the module at the time audit user role is initialized. The 48-byte random value is output from the original module onto an iKey to enable initializing the Auditor role on additional modules into the same domain. For password configurations, this value is an 8 - 255 character data string supplied by the user during configuration of the secure audit capability. **This key is a CSP.**
Secure Audit Logging Key (SALK) 256-bit HMAC key.	256-bit.	HMAC (Cert #C1718).	[SP800-90Ar1] CTR_DRBG with AES-256.	Input / Output encrypted under the RDK and using AES-256 in KWP mode.	N/A.	Flash memory in plaintext, Flash memory encrypted with RDK.	Factory Reset - KDM1	A 256-bit key used to verify data integrity and authentication of the log messages. Saved in the parameter area of Flash memory. **This key is a CSP.**
Secure Audit AccessID-HMAC Key 256-bit HMAC key.	256-bit.	HMAC (Cert #C1718).	[SP800-90Ar1] CTR_DRBG with AES-256.	Not Input or Output.	N/A.	Working SDRAM in plaintext.	Power Cycle - KDM2	A 256-bit key used to create an HMAC of the AccessID to be used in the Secure Audit logs, to prevent against the theft of the actual AccessID. A new key will be generated at every module power-on or firmware reset. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
User Password (if PED configuration and optionally selected) 8 - 255 character data string.	32 to 256-bit.	PBKDF (Cert #A480).	N/A.	Input from host using ICD communication path and encrypted under the PEC and using KTS-OAEP-basic from [SP800-56Br2].	N/A.	A salted hash of the password stored in Flash memory encrypted with PSK.	Partition deletion - KDM1	User provided password input by the operator as a second factor of authentication data. **This key is a CSP.**
AccessID Encryption Key – Encryption Key (AEK-EK) 256-bit AES key.	256-bit.	AES (Cert #C1707).	N/A..	Imported encrypted under the PEC using KTS-OAEP.	N/A.	Working SDRAM in plaintext.	Power Cycle - KDM2	A 256-bit key generated by the Thales Luna client and submitted to the HSM for use to wrap the HSM generated AEK for export to the Thales Luna Client to encrypt the AccessID. A new key is generated for each AEK transfer event. **This key is a CSP.**
AccessID Encryption Key (AEK) 256-bit AES key.	256-bit.	AES (Cert #C1707).	[SP800-90Ar1] CTR_DRBG with AES-256.	Exported encrypted under the AEK-EK using AES-KW.	N/A.	Working SDRAM in plaintext.	Power Cycle - KDM2	A 256-bit key used with AES-KWP to encrypt the AccessID. A new key will be generated at every module power-on or firmware reset. **This key is a CSP.**
AccessID 128-bit value	128-bit	N/A.	N/A.	Option 1: Imported encrypted using the AEK and AES-KW when STC is not in use Option 2: Either encrypted using AES-GCM or AES-CTR with separate HMAC and using the Partition STC Session Encryption and Authentication Keys.	N/A.	Working SDRAM in plaintext.	Power Cycle - KDM2	128-bit secret value used as an authorization token for sessions. **This key is a CSP.**
Stored User Password Hash (PED configuration) 256-bit value.	128-bit.	N/A.	SHA (Cert #C1707).	N/A.	N/A.	Flash memory encrypted with PSK.	Zeroized following use – KDM1.	Hashed user password with 256-bit random salt. SSP is compared with salted hash of passwords supplied by the end-user as part of login when using PED authentication with optional memorized secret. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
PED Authentication Data (if PED configuration) 48-byte random value.	256-bit.	KBKDF (Cert #A480).	[SP800-90Ar1] CTR_DRBG with AES-256.	Input / Output via direct connection to PED. All messages sent to the local PED are encrypted using HSM CSP Wrapping Key and AES KWP.	N/A.	Working SDRAM in plaintext (during generation).	N/A	A 256-bit random value that is generated by the module when a role is created and is written out to the iKey connected to the Thales Luna PED. **This key is a CSP.**
Password (Authentication Data if Password configuration) 8 - 255 character data string.	32 to 2040-bit.	PBKDF (Cert #A480).	N/A.	Input from host using ICD communication path and encrypted under the PEC and using KTS-OAEP-basic from [SP800-56Br2].	N/A.	Working SDRAM in plaintext (during generation).	N/A	User provided password input by the operator as authentication data. **This key is a CSP.**
User Storage Key (USK) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	[SP800-90Ar1] CTR_DRBG with AES-256.	Not Input or Output.	N/A.	Flash memory encrypted with User's Authentication Data and KEK.	Partition deletion - KDM1	This key is used to encrypt all sensitive attributes of all private objects owned by users of a partition (e.g. HSM SO, Administration, Partition Crypto Officer). **This key is a CSP.**
Partition Storage Key (PSK) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	[SP800-90Ar1] CTR_DRBG with AES-256.	Not Input or Output.	N/A.	Flash memory encrypted with USK.	Partition deletion - KDM1	This key is unique per-partition and used to encrypt all SSP that are shared by all roles of a given partition. **This key is a CSP.**
SKS Master Key (SMK) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	[SP800-90Ar1] CTR_DRBG with AES-256.	Input/Output using CPV3 or CPV4.	N/A.	Flash memory encrypted with USK.	Zeroized via ICD command - KDM1	A randomly generated 256-bit secret used as the master key for deriving all SKS key blob encryption keys. **This key is a CSP.**
HA Login Public Key (HA_{PUB}) 4096-bit public key.	150-bit.	KAS1-basic (Cert #A478, #A479, #A480, #A481).	[FIPS 186-4], Appendix B.4.1.	Certificate Output in Plaintext.	N/A.	Flash memory in plaintext.	Zeroized via ICD command - KDM1	A 4096-bit RSA public key used for the HA Login protocol. **This key is a PSP.**
HA Login Private Key (HA_{PK}) 4096-bit private key.	150-bit.	KAS1-basic (Cert #C1707).	[FIPS 186-4], Appendix B.3.6.	Not Input or Output.	N/A.	Flash memory encrypted with USK.	Zeroized via ICD command - KDM1	A 4096-bit RSA private key used for the HA Login protocol. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
HA Login Authentication Data Encryption Key PIN (RND) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	[SP800-90Ar1] CTR_DRBG with AES-256.	Output encrypted using AES-256 in KWP mode and using a shared secret from output of [SP800-56Br2], KAS1-basic exchange.	N/A.	Working SDRAM in plaintext.	Zeroized via ICD command - KDM1 Session closure - KDM1	A 256-bit encryption key used with AES to encrypt authentication data for export to the primary HA Login instance. **This key is a CSP.**
HA Login Ephemeral Wrapping Key (K_{SESS}) 256-bits AES key.	256-bit.	AES (Cert #C1707 and #C1718).	[SP800-90Ar1] CTR_DRBG with AES-256.	Output encrypted with peer TWC.	N/A.	Working SDRAM in plaintext	Zeroized via ICD command - KDM1 Session closure - KDM1	A 256-bit encryption key used with AES to encrypt authentication data for re-import from the primary HA Login instance. **This key is a CSP.**
Key Cloning Domain Vector (KCV) 256-bit key.	32 to 2040-bit.	KDA (Cert #A480).	[SP800-90Ar1] CTR_DRBG with AES-256. N/A for Password configuration.	Input / Output via direct connection to Thales PED.	N/A.	Flash Memory encrypted with PSK	Partition deletion - KDM1	Value that controls a partition's ability to participate in the cloning protocol. In the case of PED configurations, it is generated by the module or imprinted onto the module at partition initialization time. For password configurations, this 8 - 255 character data string is supplied by the user during partition initialization. For PED configurations, the 48-byte random value is output from the original partition in the domain to a PED key to enable initializing additional modules into the domain. **This key is a CSP.**
Remote PED Vector (RPV) (if PED configuration) 256-bit key.	256-bit.	KDA (Cert #A480).	[SP800-90Ar1] CTR_DRBG with AES-256.	Output via direct connection to a Luna PED.	N/A.	Flash memory encrypted with GSK	Zeroized via ICD command - KDM1 Erased on user zeroize request or destructive policy change.	A randomly generated 256-bit key, which must be shared between a remote PED and a cryptographic module in order to establish a secure communication channel between them. **This key is a CSP.**
PED Authentication Certificate (PAC) ECC public key on curve P-521.	256-bit.	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Output via direct connection to a Luna PED.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM1 Erased on user zeroize request or destructive policy change.	An ECC public key certificate used to verify certificates for local or remote connection with a Luna PED. **This key is a PSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
PED Authentication Key (PAK) ECC private key on curve P-521.	256-bit.	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM1 Erased on user zeroize request or destructive policy change.	An ECC private key used to sign certificates used for local or remote connection with the Thales Luna PED. **This key is a CSP.**
HSM Static Key-Agreement Certificate for Local Connections (HSM-SKA-C_{LOCAL}) ECC public key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Output via direct connection to a Luna PED.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM1 Erased on user zeroize request or destructive policy change.	Used by the Thales Luna PED to authenticate the local HSM to connect to and to extract the HSM's static ECC public key for C(1e,1s, ECC CDH) key-agreement for local connection with a Thales Luna PED. **This key is a PSP.**
HSM Static Key-Agreement Private Key for Local Connections (HSM-SKA-K_{LOCAL}) ECC private key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM1 Erased on user zeroize request or destructive policy change.	Used by the HSM as the static private key for C(1e,1s, ECC CDH) key-agreement agreement for local connection with a Thales Luna PED. **This key is a CSP.**
HSM Static Key-Agreement Certificate for Remote Connections (HSM-SKA-C_{REMOTE}) ECC public key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Output via direct connection to a Luna PED.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM1 Erased on user zeroize request or destructive policy change.	Used by the Thales Luna PED to authenticate the remote HSM to connect to and to extract the HSM's static ECC public key for: • C(2e,2s, ECC CDH) key-agreement for remote connection with PED. • C(1e,1s, ECC CDH) DLC Key Transport for SSP migration **This key is a PSP.**
HSM Static Key-Agreement Private Key for Remote Connections (HSM-SKA-K_{REMOTE}) ECC private key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM1 Erased on user zeroize request or destructive policy change.	Used by the remote HSM as the static private key for: • C(2e,2s, ECC CDH) key-agreement agreement for remote connection with PED. • C(1e,1s, ECC CDH) DLC Key Transport for SSP migration **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
HSM Ephemeral Key-Agreement Certificate (HSM-EKA-C) ECC public key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Output via direct connection to a Luna PED.	N/A.	Working SDRAM in plaintext	KDM1 - following use.	Used by the Thales Luna PED to authenticate the remote HSM to connect to and to extract the HSM's ephemeral public key for C(2e,2s, ECC CDH) key-agreement agreement for remote connection with a Thales Luna PED. **This key is a PSP.**
HSM Ephemeral Key-Agreement Private Key (HSM-EKA-K) ECC private key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output	N/A.	Working SDRAM in plaintext	KDM1 - following use.	Used by the Thales Luna PED to authenticate the remote HSM and to extract the HSM's ephemeral public key for C(2e,2s, ECC CDH) key-agreement agreement for remote connection with a Thales Luna PED. **This key is a CSP.**
Remote PED Vector Certificate (RPV-C) ECC public key on curve P-521.	256-bit.	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Output via direct connection to a Luna PED.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM2 or KDM1 in response to erase request via ICD command. Erased on user zeroize request or destructive policy change.	An ECC public key certificate used by the HSM device to verify PED-SKA-C, PED-EKA-C. **This key is a PSP.**
Remote PED Vector Private Key (RPV-K) ECC private key on curve P-521.	256-bit.	ECDSA (Cert #C1707 and #C1718).	[FIPS 186-4], Appendix B.4.1.	Output via direct connection to a Luna PED.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM2 or KDM1 in response to erase request via ICD command. Erased on user zeroize request or destructive policy change.	An ECC private key used by the HSM to sign PED-SKA-C, and by the Luna PED to sign PED-EKA-C. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
PED Static Key-Agreement Certificate for Remote Connections (PED-SKA-C) ECC public key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Output via direct connection to a Luna PED.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM2 or KDM1 in response to erase request via ICD command. Erased on user zeroize request or destructive policy change.	Used by the HSM to authenticate and extract the Luna PED's ECC ephemeral public key for C(2e,2s, ECC CDH) or C(1e,1s ECC CDH) key-agreement. Uniquely generated for each use. **This key is a PSP.**
PED Static Key-Agreement Private Key (PED-SKA-K) ECC private key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Output via direct connection to a Luna PED.	N/A.	Working SDRAM in plaintext	Power Cycle - KDM2 or KDM1 in response to erase request via ICD command. Erased on user zeroize request or destructive policy change.	Used by the Thales Luna PED for Remote connections. Act as An ECC static private key for C(2e,2s, ECC CDH) key-agreement. Key is not used by the HSM as a SP but is generated by it for use by the Luna PED. **This key is a CSP.**
PED Master Shared Secret 256-bit key.	256-bit.	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash (Cert #A480).	N/A.	N/A.	KAS-ECC (Cert #A480).	Working SRAM in plaintext.	Zeroized following use – KDM1.	Intermediate key value used during setup of the Local and Remote PED channel. Key is the output of the ECDH function and used to generate HSM and PED CSP Wrapping Key, MAC key, IV and Data Encryption Key. Keys are generated using OneStep KDF from [SP800-56Cr2] with SHA2-512. **This key is a CSP.**
HSM CSP Wrapping Key (CWK$_{HSM}$) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash.	Not Input or Output.	[SP800-56Ar3] fullUnified with full key validation and key-pair generation.	Working SDRAM in plaintext	PED Channel Termination - KDM1 Erased on user zeroize request or destructive policy change.	Derived during Local and Remote PED Channel for wrapping exchanged SSPs. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
PED CSP Wrapping Key (CWK_{PED}) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash.	Not Input or Output.	[SP800-56Ar3] fullUnified with full key validation and key-pair generation.	Working SDRAM in plaintext	PED Channel Termination - KDM1 Erased on user zeroize request or destructive policy change.	Derived during Local and Remote PED Channel for wrapping exchanged SSPs. **This key is a CSP.**
HSM Data Encryption Key (DEK_{HSM}) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash.	Not Input or Output.	[SP800-56Ar3] fullUnified with full key validation and key-pair generation.	Working SDRAM in plaintext	PED Channel Termination - KDM1 Erased on user zeroize request or destructive policy change.	Derived during Remote PED Channel for encrypting communication messages (from HSM-to-PED). **This key is a CSP.**
HSM MAC Key (DMK_{HSM}) 256-bit HMAC key.	256-bit.	HMAC (#C1707).	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash.	Not Input or Output.	[SP800-56Ar3] fullUnified with full key validation and key-pair generation.	Working SDRAM in plaintext	PED Channel Termination - KDM1 Erased on user zeroize request or destructive policy change.	Derived during Remote PED Channel for message authentication of communication messages (from HSM-to-PED). **This key is a CSP.**
HSM Initialization Vector (IV_{HSM}) 256-bit IV.	256-bit.	AES (Cert #C1707 and #C1718).	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash.	Not Input or Output.	[SP800-56Ar3] fullUnified with full key validation and key-pair generation.	Working SDRAM in plaintext	PED Channel Termination - KDM1 Erased on user zeroize request or destructive policy change.	Derived during Remote PED Channel as the initialization vector for encrypting communication messages (from HSM-to-PED). **This key is a CSP.**
PED Data Encryption Key (DEK_{PED}) 256-bit AES key.	256-bit.	AES (Cert #C1707 and #C1718).	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash.	Not Input or Output.	[SP800-56Ar3] fullUnified with full key validation and key-pair generation.	Working SDRAM in plaintext.	PED Channel Termination - KDM1 Erased on user zeroize request or destructive policy change.	Derived during Remote PED Channel for encrypting communication messages (from PED-to-HSM). **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
PED MAC Key (DMK_{PED}) 256-bit HMAC key.	256-bit.	HMAC (#C1707).	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash.	Not Input or Output.	[SP800-56Ar3] fullUnified with full key validation and key-pair generation.	Working SRAM in plaintext.	PED Channel Termination - KDM1 Erased on user zeroize request or destructive policy change.	Derived during Remote PED Channel for message authentication of communication messages (from PED-to-HSM). **This key is a CSP.**
PED Initialization Vector (IV_{PED}) 256-bit IV.	256-bit.	AES (Cert #C1707 and #C1718).	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash.	Not Input or Output.	[SP800-56Ar3] fullUnified with full key validation and key-pair generation.	Working RAM in plaintext.	PED Channel Termination - KDM1 Erased on user zeroize request or destructive policy change.	Derived during Remote PED Channel as the initialization vector for encrypting communication messages (from PED-to-HSM). **This key is a CSP.**
Password Encryption Key (PEK) 4096 bit private key	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Not Input or Output.	N/A.	Working RAM in plaintext.	Power Cycle - KDM1 Erased on user zeroize request or destructive policy change.	A 4096-bit RSA private key used to decrypt user passwords that are provided to the module. It is generated the first time it is required. **This key is a CSP.**
Password Encryption Certificate (PEC) 4096-bit public key certificate	150-bit.	RSA (#C1707, #C1717, #C1718 and #C1719).	[FIPS 186-4], Appendix B.3.6.	Certificate Output in Plaintext.	N/A.	Working RAM in plaintext.	Power Cycle - KDM1 Zeroized via ICD command - KDM1.	The X.509 public key certificate corresponding to the PEK. It is created and signed by the HOK the first it is required. **This key is a PSP.**
FM SMFS Encryption Key ($SMFSK_{ENC}$) 256-bit AES key.	256-bit.	AES (Cert #C1707).	[SP800-90Ar1] CTR_DRBG with AES-256.	Not Input or Output	N/A.	Flash memory encrypted with PSK or TVK (when SMFS auto-activation enabled).	Power Cycle - KDM1 Zeroized via ICD command. - KDM1.	This key is used to encrypt data submitted by an FM to the SMFS for storage. **This key is a CSP.**
FM SMFS MAC Key ($SMFSK_{INT}$) 256-bit AES key.	256-bit.	HMAC (#C1707).	[SP800-90Ar1] CTR_DRBG with AES-256.	Not Input or Output	N/A.	Flash memory encrypted with PSK or TVK (when SMFS auto-activation enabled).	Power Cycle - KDM1 Zeroized via ICD command. - KDM1.	This key is used to generate MAC over data submitted by an FM to the SMFS for storage. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
Partition STC Static Public ID Key (STC-PID$_{PUB}$) ECC public key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Certificate Output in Plaintext.	N/A.	Flash memory in plaintext	Zeroized via ICD command - KDM1	A 521-bit ECC public key used as the partition's static ID in the STC protocol. **This key is a PSP.**
Partition STC Static Private ID Key (STC-PID$_{PRIV}$) ECC private key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output.	N/A.	Flash memory encrypted with GSK	Zeroized via ICD command - KDM1	A 521-bit ECC private key used as the partition's static ID in the STC protocol. **This key is a CSP.**
Partition STC Ephemeral Public Key (STC-PKA$_{PUB}$) ECC public key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Certificate Output in Plaintext.	N/A.	Working RAM in plaintext	Zeroized via ICD command - KDM1 Session closure - KDM1	A 521-bit ECC public key used as the partition's ephemeral key in the STC protocol. **This key is a PSP.**
Partition STC Ephemeral Private Key (STC-PKA$_{PRIV}$) ECC private key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	[FIPS 186-4], Appendix B.4.1.	Not Input or Output.	N/A.	Working RAM in plaintext	Zeroized via ICD command - KDM1 Session closure - KDM1	A 521-bit ECC private key used as the partition's ephemeral key in the STC protocol. **This key is a CSP.**
Client STC Static Public ID Key (STC-CID$_{PUB}$) ECC public key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	N/A (user imported)	Input in Plaintext.	N/A.	Flash memory in plaintext	Zeroized via ICD command - KDM1	A 521-bit ECC public key used as the client's static ID in the STC protocol. **This key is a PSP.**
Client STC Ephemeral Public Key (STC-CKA$_{PUB}$) ECC public key on curve P-521.	256-bit.	KAS-ECC (Cert #A480).	N/A (user imported)	Input in Plaintext.	N/A.	Working RAM in plaintext	Zeroized via ICD command - KDM1 Session closure – KDM1	A 521-bit ECC public key used as the client's ephemeral key in the STC protocol. **This key is a PSP.**
STC Master Shared Secret 256-bits.	256-bit	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash (Cert #A480)).	N/A.	N/A.	KAS-ECC (Cert #A480).	Working RAM in plaintext	KDM1 - following use.	Intermediate key value during the STC key negotiation. Key is the output of the ECDH function and used to generate the Partition STC Session Encryption and Authentication Keys using OneStep KDF from [SP800-56Cr2] with SHA2-512. **This key is a CSP.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
Partition STC Session Encryption and Authentication Keys (STC-PEN, STC-PMA, STC-PIV, STC-CEN, STC-CMA, STC-CIV).	256-bit	AES (Cert #C1707 and #C1718). HMAC (Cert #C1707).	OneStep KDF from [SP800-56Cr2] and using SHA2-512 as hash.	Not Input or Output.	[SP800-56Ar3] fullUnified with full key validation and key pair generation.	Working RAM in plaintext	Zeroized via ICD command - KDM1 Session closure - KDM1	These keys are agreed upon with a client application for the purpose of encrypting and generate MAC for message exchanges during an STC session. **This key is a CSP.**
Asymmetric Key Pairs (general partition or session keys) RSA, DSA, ECC, DH.	112 to 256-bit for ECC keys depending on the curve. 112 to 128-bit for DSA keys depending on modulus size. 112 to 201-bits for RSA keys depending on modulus length. 112 to 150-bit for DH keys depending on modulus length.	RSA (#C1707, #C1717, #C1718 and #C1719). ECDSA (Cert #C1707 and #C1718). DSA (Cert #C1707 and #C1718). KAS-ECC-SSC (Certs #A478 and #A480). KAS-FFC-SSC (Certs #A478 and #A480). KAS-RSA (Certs #A478 and #A480). KTS-RSA (Certs #A478 and #A480).	N/A (user imported) Or [FIPS 186-4], Appendix B.4.1. – for ECC key-pair. Or [FIPS 186-4], Appendix B.3.6. – for RSA, DH and DSA keys.	Input or output encrypted using Symmetric Keys (general partition or session keys) using key wrap/unwrap ICD commands using key wrap/unwrap ICD commands and [SP800-38F] encryption options. Input using Symmetric Keys (general partition or session keys) using key unwrap ICD commands and approved symmetric algorithms as permitted by [FIPS 140-3 IG] D.G, Key transport methods. When transferred between partitions using SKS, encrypted under the SMK.	N/A.	Flash memory encrypted with USK.	Zeroized via ICD command - KDM1 Session closure – KDM1	General use asymmetric key pairs that can be exported/imported from/to the module or generated by the module. **This key is a CSP (private key component).** **This key is a PSP (public key component).**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
Symmetric Keys (general partition or session keys) AES or Triple-DES (including AES-XTS), MAC, KDF.	128, 192 or 256-bit for AES keys. 112-bit for Triple-DES.	AES (Cert #C1707 and #C1718). Triple-DES (Cert #C1707). HMAC (Cert #C1707). CMAC (Cert #C1707). KDA (#A480) KBKDF (#C1707). KTS (AES Cert #C1707 and #C1718).	N/A (user imported) Or [SP800-90Ar1] CTR_DRBG with AES-256 (module generated).	Input or output encrypted using Symmetric Keys (general partition or session keys) using key wrap/unwrap ICD commands and [SP800-38F] encryption options. Input or output encrypted using Asymmetric Keys (general partition or session keys) using key wrap/unwrap ICD commands and KTS-OAEP-basic from [SP800-56Br2]. Input using Symmetric Keys (general partition or session keys) using key unwrap ICD commands and approved symmetric algorithms as permitted by [FIPS 140-3 IG] D.G, Key transport methods. When transferred between partitions using SKS, encrypted under the SMK.	Can be established as the output of supported [SP800-56Ar3] compliant key establishment using other partition stored asymmetric key-pair.	Flash memory encrypted with USK.	Zeroized via ICD command - KDM1 Session closure – KDM1	General use symmetric key pairs that can be exported/imported from/to the module or generated by the module. **This key is a CSP.**

9.2 Non-Deterministic Random Bit Generation Specification

The module includes a non-deterministic Random Bit Generator (RBG) within the module boundary.

The non-deterministic RBG is used exclusively to feed an approved conditioning function where in-turn the output of the conditioning function is used to seed the DRBG (Cert #C1707).

The Non-Deterministic RBG complies with [SP800-90B] and has been validated using with guidance set out in [FIPS 140-3 IG] .

Table 9-2: Non-Deterministic Random Number Generation Specification

Entropy sources	Minimum number of bits of entropy	Details
Non-deterministic jitter in from FRO.	Full-entropy output	[SP800-90B] compliant Non-Deterministic RBG using a hardware based noise internal to the module boundary. Digitized output from the noise source is fed through an approved conditioning function based on SHA2-512 (Cert #C1707). Raw noise is generated based on non-deterministic jitter built up in free-running oscillators. The module achieves full entropy from the output of the conditioning function where every 384-bits used to seed the DRBG includes 384-bits of entropy. All outputs from the noise source are subjected to statistical testing ahead of being fed to the conditioning function. The output of the hardware noise source includes a total failure test to check for bit-patterns consistent with hardware failures.

9.3 Key Import/Export methods

Depending on the configuration of the module, the following methods of key import and export for ‘Asymmetric Key Pairs (general partition keys)’ and ‘Symmetric Keys (general partition keys)’ are available as a service:

> **Key Wrap / Unwrap using Cloning Protocol Version 4 (CPV4)**

CPV4 uses the following cryptography:

- ECDH OneStep Key Derivation (section 5.8.2.1 from [SP800-56Ar3]) with C(2e, 0s, ECC CDH) with either P-521 or BrainpoolP512r1 as a curve to derive the CPV4 Shared Secret;
- OneStep KDF with either SHA2-512 or SHA3-512 is used to derive all keys as part of the CPV4 protocol;
- ECDSA with either P-521 or BrainpoolP512r1 are used for signing the ephemeral CPV4 Source Public Key alongside exchanged messages;
- AES-256 in either GCM mode or in CTR mode are used to encrypt sensitive objects in transfer using the protocol. Where CTR mode is used, transferred objects independently also include either a HMAC-SHA2-512 or HMAC-SHA3-512 MAC.

Exact cryptography used is selected based on a down-selected cipher suite during the initial CPV4 negotiation.

> **Key Wrap / Unwrap using Cloning Protocol Version 3 (CPV3)**

Cloning is a product feature where KAS1-basic from [SP800-56Br2] is used to negotiate a shared secret used to transfer partition objects between a source and destination partition and where these can be on the same or different cryptographic module. The protocol uses the following options with KAS1-basic:

- RSASVE for transfer of shared secrets uses the public key from the TWC4 certificate which has a modulus length of 4096-bits;
- The TWC4 certificate used is generated by an instance of the module as a trusted third party (TTP) and is signed by the generating modules HOC. All TWC3 keys are generated using rsakpg1-crt from section 6.3.1.3 of [SP800-56Br2] and where the generating module will perform key pair validation as per steps 2 and 3b from section 6.4.1.1 as the TTP. On receipt of a destination modules TWC4 used during the CPV3 protocol, the source module will validate the certificate and its associated certificate chain back to a shared common root certificate. This establishes the originating HSM as a TTP as defined in [SP800-56Br2].
- Shared keys are derived using One-Step KDF from [SP800-56Cr2] using SHA2-512. Inputs to the KDF include the exchanged shared secret from the RSASVE transfer, alongside the pre-shared 256-bit secret key (KCV or RDK) and additional HSM related shared information; and
- Encryption of the SMK during the transfer uses AES-256 in KWP mode and a single-use key and IV derived from the output of the KDF.

This scheme uses a hybrid key transport method compliant with section 9.3 of [SP800-56Br2] where KAS1-basic is used as a key establishment scheme followed by use of AES-256 in KWP mode as a [SP800-38F] compliant key wrapping algorithm.

> **Scalable Key Storage (SKS)**

SKS allows the transfer of partition objects (symmetric and asymmetric keys, alongside other objects) between partitions encrypted under the SMK, which must have been pre-shared between source and destination partition using either CPV3 or CPV4.

SKS uses AES-256 in GCM mode with a 128-bit random IV generated by the cryptographic module using output from its [SP800-90Ar1] DRBG, and where a unique key per extraction is used. This key is deriver using the shared partition SMK and a 256-bit random salt value (unique per SKS export operation) and the SMK.

Encryption keys are derived using [SP800-108r1] PRF KDF and using AES-CMAC-256.

> **Key Wrap / Unwrap**

The key wrap operation is available for use to import or export raw Symmetric Keys (general partition or session keys) or an Asymmetric Key Pair (general partition or session keys) – private key, using one of the following options:

- KTS-OAEP-basic from [SP800-56Br2] and where the following options are supported:
  - Modulus lengths of 2048, 3072, 4096, 6144, or 8192 for export or 1024, 2048, 3072, 4096, 6144, or 8192 for import; and
  - Hash and MGF options must match and be consistent with one of the following algorithms: SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.
  - Where this mechanism is used with public keys generated by the module these are generated using either rsakpg1-crt or rsakpg2-crt from section 6.3.1.3 of [SP800-56Br2] and where the module is the ‘owner’. All keys generated by the module are subject to a pairwise consistency check on generation and are separately validated as per either section 6.4.1.2 or 6.4.1.3 from [SP800-56Br2] depending on the generation method used.
  - Where this mechanism is used with a public key imported from outside the module, assurances as per section 6.4.2 of [SP800-56Br2] shall be sought ahead of use of the public key. In this scenario, the module is not providing any assurances for the generation methods of the

public/private key pair and where the resulting encrypt operation is not considered part of a key transport scheme as defined in [FIPS 140-3 IG], D.G.

- [SP800-38F] compliant KTS using one of the following options for both key unwrapping and wrapping:
  - AES (128, 192 or 256-bit) in KW, KWP.
- [FIPS 140-3 IG] D.G, Key transport methods, compliant KTS for unwrap of key objects using one of the following options:
  - AES (128, 192 or 256-bit) in CBC, CTR or ECB modes; or
  - Triple-DES (112 and 168-bit) in CBC, ECB and CTR.

The unwrap operation takes as input an encrypted symmetric key or asymmetric private key and a handle to the key required to successfully unwrap the object. It decrypts the key and returns the handle to the imported key.

> **Key Unwrap using historic versions of SKS**

The module supports key import for keys previously exported from another certified Thales HSM using versions of SKS supported by legacy firmware versions, and where related certificates are now on NIST's historical list.

Import is supported for key migration only.

Objects imported using this method are either:

- encrypted using AES with 256-bit key in GCM mode with SHA2-256 for integrity protection; or
- encrypted using AES with 256-bit key in OFB mode and with SHA1 for integrity protection.

NOTE Where a key is generated by the module using a generation or derivation method in a non-approved mode of operation and exported, this SSP shall not be re-imported to the module and used when configured in its approved mode of operation.

10 Self-Tests

10.1Pre-Operational tests

The module performs the pre-operational self-tests upon power-up to confirm the firmware integrity, and to check the continued correct operation of the random number generator and each of the implemented cryptographic algorithms used in support of the integrity checks.

While the module is running these self-tests, all interfaces are disabled until the successful completion of the self-tests. If any test fails an error message is output alongside being recorded in the error log, the module halts, and data output is inhibited.

Table 10-1: Pre-operational self-tests

Test	Operations Performed	Indicator
SHA (SHA-1 and SHA2-384) KAT.	Digest.	Error output and module halt.
RSA (4096-bit modulus) KAT.	Sign and Verify.	Error output and module halt.
Boot loader performs an RSA PKCS #1-v1.5 signature with 4096-bit modulus and SHA2-384 signature verification of itself.	Verify and Digest.	Error output and module halt.
Boot loader performs an RSA PKCS #1-v1.5 signature with 4096-bit modulus and SHA2-384 signature verification of the main firmware image.	Verify and Digest.	Error output and module halt.

> **NOTE** Signature verification pre-operational self-tests will always be preceded by the Conditional KAT on the bootloader implementations of RSA supporting a single mode of operation. Transition from an approved to non-approved mode of operation automatically triggers the HSM zeroize or decommission module service.

10.2 Conditional tests

The module automatically performs conditional self-tests based on the module operation. These self-tests do not require operator input to initiate.

> **NOTE** When conditional tests are run as part of the pre-operational self-test, the HSM will test all possible implementations of a given algorithm independent of the HSM level configuration and settings.
>
> During PST, the module will exclusively test the implementation of a given algorithm in use for a given configuration and settings of the HSM at the time of a given conditional test executing,

Implemented conditional tests are in one of the following forms:

> Known Answer Test (KAT);

> Pair-wise Consistency Test (PCT);

> Statistical testing; or

> Hardware failure testing.

All KAT alongside statistical testing of the noise source is performed immediately following the pre-operational self-test at module power-on.

Table 10-2: Conditional self-tests (Firmware)

Test	Cryptographic Mechanism Tested	Location	When Performed	Operations Performed	Indicator
SHA KAT.	**Pre-operational:** SHA-1 and SHA2-384. **PST:** N/A.	Bootloader.	Prior to first use.	Digest.	Error output and module halt.
RSA KAT.	**Pre-operational:** RSA PKCS #1-v1.5, modulus 4096, SHA2-384. **PST:** N/A.	Bootloader.	Prior to first use.	Sign and Verify.	Error output and module halt.
HRBG conditional tests.	**Continuous Test:** Total failure test on the output from the hardware noise source, Repetition Count Test and Adaptive Proportion Test statistical tests.	Firmware.	Continuous.	N/A.	Error output and module halt.
Firmware Load Test	**Continuous Test:** RSA PKCS #1-v1.5, modulus 4096 signature and SHA2-384.	Firmware	On firmware update request.	Verify.	Error output and FW update request rejected.
RSA PCT.	Performed for all RSA key generation mechanism.	Firmware.	On generation.	Encrypt, Decrypt, Sign and Verify.	Error output and module halt.
DSA PCT.	Performed for all DSA key generation mechanism.	Firmware.	On generation.	Sign and Verify.	Error output and module halt.
ECC PCT (covers keys used for ECDSA and ECDH).	Performed for all ECC key generation mechanism.	Firmware.	On generation.	Sign, Verify and Derive.	Error output and module halt.
DRBG KAT.	**Pre-operational:** Instantiate, Generate and Reseed KAT for CTR_DRBG with AES-256. **PST:** *<as per pre-operational self-test.>*	Firmware.	Prior to first use, PST.	N/A.	Error output and module halt.
SHA KAT.	**Pre-operational:** SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE-128, SHAKE-256. **PST:** hash are tested based on inclusion in the KAT for higher-order algorithms.	Firmware.	Prior to first use, PST.	Digest.	Error output and module halt.

Test	Cryptographic Mechanism Tested	Location	When Performed	Operations Performed	Indicator
HMAC KAT (General).	**Pre-operational:** HMAC-SHA1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512. **PST:** HMAC-SHA1, HMAC-SHA2-224, HMAC-SHA2-384, SHA3-256.	Firmware.	Prior to first use, PST.	Digest.	Error output and module halt.
HMAC KAT (Secure log implementation).	**Pre-operational:** HMAC-SHA2-256. **PST:** HMAC-SHA2-256.	Firmware	Prior to first use, PST.	Digest.	Error output and module halt.
RSA KAT[47].	**Pre-operational:** Signature Generation, Sig Verification for RSA X9.31 with SHA2-256, RSA PKCS #1-v1.5 with SHA2-256, RSA PKCS #1-v1.5 (no hash), RSA PKCS #1-v1.5 PSS with SHA2-256 and SHA2-256 for MGF, [SP800-56Br2] RSA-OAEP-basic with 2048-bit modulus and SHA1, SHA2-256 and SHA2-384, SHA2-512 as MGF. **PST:** Signature Generation, Signature Verification for: PKCS-PSS with modulus of 8192-bits and SHA2-256, PKCS-PSS with modulus of 2048-bit and SHA2-256, RSA-OAEP-basic with 2048-bit modulus and SHA2-256 as MGF.	Firmware.	Prior to first use, PST.	Sign, Verify, Encrypt and Decrypt.	Error output and module halt.
DSA KAT[48]	**Pre-operational:** Signature Generation, Signature Verification for 2048-bit modulus with SHA2-224. Signature Verification with 1024-bit modulus and SHA-1. **PST:** Signature Generation, Signature Verification for 2048-bit modulus with SHA2-224.	Firmware.	Prior to first use, PST.	Sign and Verify.	Error output and module halt.
Diffie-Hellman KAT.	**Pre-operational:** X9.42 Diffie-Hellman [SP800-56Ar3] key derive with 2048-bit modulus. **PST:** *<as per pre-operational self-test.>*	Firmware.	Prior to first use, PST.	Derive.	Error output and module halt.

[47] random values used during the KAT operation of the PKCS-PSS are fixed as per [FIPS 140-3 IG], 10.3.A with both signature generation and signature verification operations tested independently.

[48] random values used during the KAT operation are fixed as per [FIPS 140-3 IG], 10.3.A with both signature generation and signature verification operations tested independently.

Test	Cryptographic Mechanism Tested	Location	When Performed	Operations Performed	Indicator
AES KAT.	**Pre-operational:** ECB, CBC, OFB, CFB128, CFB8, KW, KWP, GCM, XTS and CMAC covering 128-bit,192-bit and 256-bit keys as supported by the different modes. **PST:** CBC, GCM, KWP (data object <=2KB), KWP (data object >2KB) – 256-bit key.	Firmware.	Prior to first use, PST.	Encrypt and Decrypt.	Error output and module halt.
Triple-DES KAT.	**Pre-operational:** ECB, CBC, OFB, CFB64, CTR, CMAC for 168-bit keys. **PST:** ECB with 168-bit key, CMAC with 168-bit key.	Firmware.	Prior to first use, PST.	Encrypt and Decrypt.	Error output and module halt.
ECDH KAT.	**Pre-operational:** ECC CDH [SP800-56Ar3] shared secret calculation (only) using curves P224, P384, P521 and K233. **PST:** ECC CDH with P384, OneStep KDF using SHA2-256.	Firmware.	Prior to first use, PST.	Derive.	Error output and module halt.
ECDSA KAT[49].	**Pre-operational:** Signature Generation, Signature Verification with ECDSA using curves P256 and K233 (no hashing). **PST:** Signature Generation, Signature Verification using ECDSA using curves P256 and K233 (no hashing).	Firmware.	Prior to first use, PST.	Sign and Verify.	Error output and module halt.
KBKDF KAT.	**Pre-operational:** KBKDF [SP800-108r1] with AES-CMAC, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512 as PRF options. **PST:** KBKDF [SP800-108r1] using AES-CMAC as the PRD with 128-bit key.	Firmware.	Prior to first use, PST.	Derive.	Error output and module halt.

[49] random values used during the KAT operation are fixed as per [FIPS 140-3 IG], 10.3.A with both signature generation and signature verification operations tested independently.

Test	Cryptographic Mechanism Tested	Location	When Performed	Operations Performed	Indicator
KDF KAT.	**Pre-operational:** OneStep KDF [SP800-56Cr2] with SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. X9.42/X9.63 KDF using SHA1. **PST:** OneStep KDF [SP800-56Cr2] with SHA2-512. X9.42/X9.63 KDF using SHA1.	Firmware.	Prior to first use, PST.	Derive.	Error output and module halt.
KAS1-basic KAT.	**Pre-operational:** KAS1-basic [SP800-56Br2] with 4096-bit modulus. **PST:** *<as per pre-operational self-test.>*	Firmware.	Prior to first use, PST.	Encrypt and Decrypt.	Error output and module halt.
PBKDF KAT.	**Pre-Operational** PBKDF [SP800-132] using HMAC-SHA1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512. **PST:** PBKDF [SP800-132] using HMAC-SHA2-512.	Firmware.	Prior to first use, PST.	Derive.	Error output and module halt.

10.3 Periodic Self-Tests

The module will perform periodic self-tests (PST) at set intervals of time for the pre-operational tests and a subset of the KAT tests.

These tests will be performed every 24 hours at which point the PSTs will be implemented as a single asynchronous command with multiple steps that make up all PSTs that must be executed. The command will be added to the HSM's command scheduler run queue, alongside any other commands that have been sent to the HSM.

Each time the PST command is given time to execute, it will perform a single step and then return priority to other commands in the queue. Each step will be consistent in size with other cryptographic commands so as not to impact overall performance of the HSM.

Conditional tests performed periodically are identified in Table 10-2 above as tests with 'PST' in the 'When Performed' column.

11 Life-cycle Assurance

11.1 Choosing a secure location for the module

Thales Luna K7 Cryptographic Module should be deployed in a secure environment that will protect the module from sophisticated attackers with direct access.

This is standard practice for high-value assets such as HSMs and forms part of a defense-in-depth approach to security.

Securing the environment of the HSM typically will include a combination of both:

- securing its location using physical defenses; and
- procedures for monitoring and managing authorized access to the HSM.

The exact measures put in place will vary and should be commensurate with the potential consequences or costs associated with the complete compromise of the HSM and cryptographic keys (or data objects) it protects.

Common components of a physical security solution often include:

- dedicated areas (e.g. locked cage or cabinet) for the HSM as part of a general IT environment;
- monitored and audited physical access controls on IT environments hosting the HSM;
- hardened locks, doors and walls to increase the effort required to force access to the HSM;
- out-of-hours alarm systems on areas containing the HSM;
- 24hr/365day on-site or remote guard service that will respond to alarms; and
- CCTV monitoring of areas containing the HSM to allow detection of activity in proximity to the HSM.

11.2 Performing secure initialization of the HSM

Ahead of using the module it must be initialized, after which it should be immediately configured into its approved mode of operation. Prior to secure initialization of the module, access control relies on procedural controls only and where the module should be received in the zeroised state with no initialized roles.

> **NOTE** The module shall be received in a zeroised state. To check the status of the module use the `hsm showinfo` LunaCM command as described in section 11.12, ‘Checking Module Status’.
>
> The module is confirmed as being in the zeroised state is the `partition status` for the administration slot reports `zeroized`.

Initialized creates the HSM SO role, names the module and associates the admin partition with a key cloning domain.

Initialization is performed using the `hsm init` command from LunaCM for Thales Luna PCIe HSM or LunaSH for Thales Luna Network HSM.

It should be noted that the `hsm init` command should only be run when an individual has been assigned to the HSM SO role and usually is run either by them or with them present.

Following initialization of the module, it should immediately be configured into its approved mode of operation ahead of initialization of any further roles or creation of any stored key objects. Guidance on configuring the approved mode of operation is provided in section 13.2.

> **NOTE** As part of initialization when using PED based authentication, the end-user is asked if they wish to duplicate your iKey. It is strongly recommended that you do this and for duplicate keys to be retained in secure storage for backup purposes. It is not possible to copy iKey at a later point.

11.3 Protection of data outside the HSM

Security of the overall system including the HSM is only as strong as its weakest component. As such – the environment needs to take responsibility for securing artefacts relating to the HSM when outside its control. In particular, the following explicit requirements shall be met:

- Where the Scalable Key Storage (SKS) Master Key (SMK) is transferred to multiple HSMs, all HSMs must be deployed to an environment that meets the minimum security requirements applicable to a given deployment as appropriate and derived from guidance provided in section 11.1.
- Audit logs extracted from the HSM should have their confidentiality protected (as appropriate) during storage outside the HSM and should be stored in a way to minimize loss of individual log records that could lead to false positives in relation to log integrity verification failure during log parsing activities.
- Secret data stored on iKey shall be protected at all times (where PED authentication is in use) – this includes authentication iKeys alongside iKeys used to transfer and store plaintext secret such as KCV, RDK and RPV.
- exported encrypted copies of the SALK shall be protected at all times. The SALK is encrypted under the RDK but as a long-term key, the protected of the encrypted SALK during transfer between HSM mitigates any risk associated with compromise of the plaintext RDK.
- Access to external encrypted key storage should be maintained on a ‘need to know’ basis in order to minimize un-necessary creation of copies of the encrypted key objects. This step is recommended to minimize potential future risks should the cryptographic algorithms used to protect keys experience a reduction in their security strength based on advances in cryptology.
- Secret keys or passwords used for authorization ahead of key use that are stored outside the HSM should be encrypted even in a system within the supported IT environment.
- Authentication credentials (including AccessID that have authenticated session) must be protection from disclosure beyond users with the associated privileges to use them. In particular this should include:
  - Prohibiting use of client applications from sources that cannot be trusted with access to authenticated sessions if either login is performed through them OR an existing AccessID is shared with them.

In addition to the above, to minimize risks to data being transferred in plaintext through the local environment of the HSM:

- All sensitive data should be encrypted if stored outside the HSM as part of a strategy to keep sensitive data logically separate from other data managed by the IT environment.
- Where possible, logically-independent ports should be used for data ingress and egress to the server hosting the HSM.
- All physical and logical connections to Trusted IT system hosting the HSM should be controlled to prohibit attempts to eavesdrop or modify sensitive traffic.
- No peripheral devices should be connected to the USB port other than authorized Thales devices. In particular, no networking devices (wireless of wired) should be attached to this interface.

Permitted devices at the time of writing this document include: Thales Luna PED and the USB-to-Serial dongle supplied for use with functionality modules.

11.4 Reviewing the Module’s log

The HSM maintains a host accessible log of events in PCIe accessible FRAM memory. This allows the log on the Thales Luna K7 Cryptographic Module to be read by the host driver even if the bootloader or firmware has failed during power-on leaving the card in an un-responsive state.

> The FRAM log can viewed using the `lunadiag` tool installed with the Thales Luna client:

- to view the FRAM log select option 18 Read Diagnostic Log then one of:
  - option 3 `Tamper` – will output event in the log dedicated to tamper with recorded events.
  - option 1 `card history` – records reset events that can be correlated with either when a card has been turned on or a soft or hard reset has had to occur.

The following figures show example output from the logs:

Tamper Log

```
Entry    0, 0x25 bytes read, timestamp = 7630, reset count = 1:

New FRAM LOG created.

Entry    1, 0xc9 bytes read, timestamp = 17167, reset count = 4:

LOG(TAMPER):   ds3644_initialize - Warning: 10 L3 tamper settings do not match
default! Registers re-configured.

Actual values: 0x8f 0xff 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x5c 0xd2 0x7e

Entry    2, 0x31 bytes read, timestamp = 17167, reset count = 4:

ALM2008: Internal data corruption
```

Card History Log

```
Entry    0, 0x20 bytes read, timestamp = 303, reset count = 1965:
reset reason 0x9154008 Date: 2019-05-30 (Thursday) Time:  1:09:09
Entry    1, 0x20 bytes read, timestamp = 304, reset count = 1966:
reset reason 0x9154008 Date: 2019-05-30 (Thursday) Time:  1:09:38
Entry    2, 0x20 bytes read, timestamp = 3097, reset count = 1967:
reset reason 0x41154008 Date: 2019-05-30 (Thursday) Time: 22:06:07
```

When reading the FRAM log messages:

> `timestamp` – is the elapsed time in milliseconds since the last time the card was reset; and

> `reset count` – identifies the reset event the offset is relevant to. This can be used with the card history log to calculate the time an event occurred by taking the data and time listed against the reset event and then adding the offset from the timestamp.

In the scenario of a power-on or on-demand self-test failing, even if the triggered event causes the module to enter the halted state, the details can be accessed through the FRAM log. If the FRAM logs cannot be read following halt, this likely indicates the power-on self-tests detected an error during startup of the bootloader. If the firmware experienced a halt during its power-on tests, then further error details will still be accessible through the FRAM logs. An example of the module failing the SHA224 self-test on startup is below:

```
---Entry  155, 0x40 bytes read, timestamp = 6495, reset count = 117:
LOG(CRITICAL): SHA2_SelfTest failed, rc=0x30000a
```

11.5 Protecting Authentication and Authorization Data

In order to maintain the separation of user roles throughout the life of the HSM deployment and to avoid compromise of a role, end users MUST:

- securely store authentication iKeys (where used) at all times;
- avoid (where used) storing corresponding PIN alongside authentication iKeys;
- never lend iKeys and/or disclose challenge-secret, PIN or passwords to anyone including other authorized end users of the HSM;
- always inspect authentication iKey (where used) prior to use to check for any signs of possible tamper; and
- avoid writing down challenge-secrets, PIN or passwords in plaintext form and/or ensure any printed or written copies of passwords are either separately encrypted or stored in a secure container only accessible to the owner of the password or challenge-secret.

Should the end user fail to comply with these requirements this could lead to subsequent compromise or malicious misuse of the HSM and its cryptographic keys.

> ⚠ **CAUTION!** Should the end user fail to comply with these requirements this could lead to subsequent compromise or malicious misuse of the HSM and its cryptographic keys.

> ⚠ **CAUTION!** In order to securely use the Thales Luna PED in its remote configuration, it is important to check and acknowledge the serial number of the target HSM during setup of the Remote PED tunnel.
>
> If the displayed serial number does not match the expected target HSM serial number the user must reject the displayed serial number at the PED, which will halt channel setup.

11.6 Managing Lost or Stolen iKey

11.6.1 User Authentication Tokens

Should an end user lose iKey or believe their iKey to have been compromised it is imperative for the security of the HSM deployment that immediate action is taken to:

1. minimize the chances of subsequent misuse of the lost or compromised iKey; and
2. check for evidence of misuse of the iKey to allow for wider compromise recovery actions to be considered.

Following identification of a lost or compromised iKey, the following actions should be taken:

- If a backup token was made and it includes a corresponding PIN, duplicate the iKey to allow re-issue (while retaining a backup) but ensure the PIN is changed on all residual copies of the duplicated iKey prior to re-deployment of the iKey.
- If the iKey was originally issued with no PIN, the iKey should be considered compromised and will need to be recreated:
  - Recreating an iKey requires all objects stored in the impacted partition or HSM to be backed up to allow for recovery following recreation of a new token
    - Where supported, cloning and SKS should be used to back up the HSM contents to a secondary HSM.
    - If an HSM SO iKey has been lost, all partitions and objects on the HSM should be backed up if partitions are still available.
    - If a Partition CO, Partition LCO or Partition CU iKey is lost, the impacted partition should be backed up.

  - If the AU token is lost, a copy of the Audit Logging Secret should be made and transferred to an iKey.
  - For the loss of an HSM SO token, LunaSH or LunaCM commands `hsm factoryReset` followed by `hsm init` should be run to re-create the iKey. This will result in a loss of all un-backed up objects on the HSM.
- In the event of loss of a Partition CO, Partition LCO or Partition CU token, the HSM SO must use the `partition delete` command to remove a partition before subsequently creating a new partition using `partition create.`
- For AU, the audit partition must be re-initialised using the LunaSH command `audit init` . Run `role init -name audit` in LunaCM to recreate a new AU token. The Audit Logging Secret can then be imported using the `audit secret import` from LunaSH or `audit import` from LunaCM.

11.7 Managing Lost or Stolen Passwords

11.7.1 General

Should a role believe their PIN or password to have been compromised (where used) but access to the corresponding token or to the HSM was not possible, the following action should be taken:

> the PIN or password should be changed using:

- `role changepw` LunaCM command for a compromised password or iKey PIN to issue a new password and the option to create a new iKey PIN where appropriate.

NOTE `role resetpw` is a sister command to `role changepw` and when enabled as a capability can enable a Security Officer to reset a lost or forgotten password on behalf of an end user.

At this time it is not possible to reset a password or PIN associated with the HSM SO role and as such, the only route to change this PIN is to back up and then to reinitialize the HSM.

11.7.2 KCV

The KCV is used to register an HSM with a domain allowing it to transfer keys with other modules. Loss of a domain key should be considered a security event.

Backups of the KCV or print-outs (optional) of the HSM-or-Partition domain secret should be retained. If this has not been performed, it is not possible to recreate or replace the domain secret.

In order to recover from complete loss of a domain secret, the objects in the HSM (where configuration permits) need to be exported and re-imported into an HSM registered with a new domain.

11.7.3 Remote PED iKeys

When a Remote PED iKey is considered to be compromised, a new iKey should be generated and distributed to all remote PED on the Orange iKey.

In order to create a new Remote PED iKey:

> Thales Luna PCIe HSM:

- run the `ped vector init` from LunaCM;

> Thales Luna Network HSM:

- run the `ped vector init` from LunaCM or `hsm ped vector init` from LunaSH.

11.8 Revoking Roles

When an individual no longer has the requirement to hold the authorized role associated with the HSM, a hand-over of iKeys and corresponding PIN or password should be arranged.

When a token has been lost for a role to be revoked, guidance on recovering from a lost end user authentication iKey in section 11.6 should be followed.

11.9 Key deletion

Keys can be deleted from a partition in one of a number of ways:

- decommissioning the module as covered in section 11.10;
- deleting the partition using the `partition delete` LunaCM command as the HSM SO;
- calling in the `C_DestroyObject` Cryptoki API command that lets a Partition CO delete any partition object owned by them;
- zeroization in response to authentication failure events (e.g. the HSM SO exceeding failed login threshold for the HSM zeroizes the entire HSM; the Partition SO exceeding failed login threshold for a user partition will zeroize the partition); and
- the entire module flash is erased using the bootloader `terase` and `tplease` commands. This erases the firmware (excluding bootloader) and all keys on the module.

> **NOTE** Use of the `terase` and `tplease` bootloader commands to perform a complete erase of all Flash based storage is not intended to be performed by customers and is included here for completeness only.
>
> The Flash contains keys created during manufacture that cannot be replaced without repeating the full manufacturing process for the card.
>
> Following erase of the flash, only signed firmware in a format not made publicly available can be loaded onto the module.

11.10 Decommissioning the HSM

Decommissioning is the process of removing all sensitive information from the cryptographic module and/or the appliance hosting it. The following sections cover the steps required in order to decommission.

11.10.1 Thales Luna PCIe HSM

The Thales Luna PCIe HSM is equipped with a two-pin decommission input, as covered in section 3.1.

Short-circuiting the decommission jumper header decommissions the HSM. You can use the blade of a screwdriver, or other electrical conducting tool to short-circuit the two pins of the decommission header, or you can connect a switch to the decommission header if desired. Power is not required to decommission the HSM. That is, you can decommission the HSM after removing it from the chassis.

When you decommission a Thales Luna PCIe HSM, the HSM is zeroised, all user accounts are deleted, and the HSM is returned to its factory state. Any firmware or partition upgrade licenses installed on the HSM are retained.

Following decommission of the HSM using this method, confirm the operation was successful using the `hsm showinfo` LunaCM command on the next power-cycle. The below text response should be shown:

```
Manually Zeroized: Yes
```

This confirms the decommission process was executed successfully.

Alternatively, run the LunaCM `hsm factoryreset` command to decommission the HSM to factory default settings. Care should be taken to observe that the command executes to a successful completion.

An example output from a successful factory reset is shown below:

```
lunacm:>hsm factoryreset
You are about to factory reset the HSM.
All contents of the HSM will be destroyed.
HSM policies will be reset and the remote PED vector will be erased.
Are you sure you wish to continue?
Type 'proceed' to continue, or 'quit' to quit now ->proceed
Command Result : No Error
```

Figure 11-1: Example successful factory reset console output from LunaCM

11.10.2 Thales Luna Network HSM

For full decommission (removing the unit from service, clearing the HSM of all your material, clearing the appliance of all identifying information) of a functioning Thales Luna Network HSM, follow these steps in LunaSH, using the serial connection:

1. Rotate all logs:

   ```
   lunash:> syslog rotate
   ```

2. Delete all files in the SCP directory:

   ```
   lunash:> my file clear
   ```

3. Delete all logs:

   ```
   lunash:> syslog cleanup
   ```

4. Return the appliance to factory-default settings:

   ```
   lunash:> sysconf config factoryreset -service all
   ```

5. Delete any backups of settings:

   ```
   lunash:> sysconf config clear
   ```

6. Push the decommission button (small red button, inset in the Thales Luna Network HSM back panel). The switch is tactile and has a travel of 1.4mm.

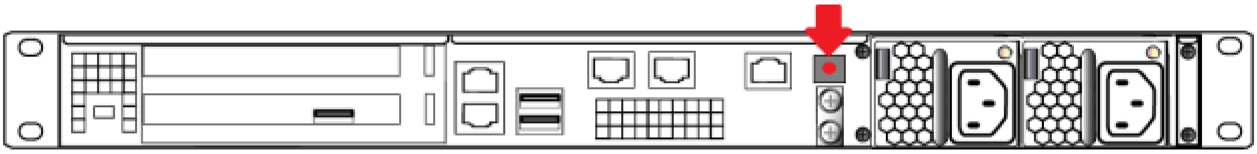

Figure 11-2: Thales Luna Network HSM decommission button location

7. Power down the appliance.

8. Power up the appliance. At this point, the HSM internally issues and executes its zeroise routine to erase all partitions and objects. This step takes about five minutes. The KEK is already gone at that point – erased as soon as the button is pressed – so the step of erasing partitions and objects is for customers subject to especially rigid decommission protocols.

11.11 Updating Firmware

Updating the module's firmware requires the HSM SO and the firmware update file to complete.

Run the LunaCM `hsm updatefw` command to update the current firmware to a new version. If any failures are detected during the update the command will fail and the module will continue running on the existing firmware.

An example output from a successful firmware update is shown below:

```
lunacm:>hsm updatefw -fuf fwupdateK7_testCert_7.0.1_RC327.fuf -
authcode fwupdateK7_testCert_7.0.1_RC327.fuf.txt
        You are about to update the firmware.
        The HSM will be reset.
        Are you sure you wish to continue?
        Type 'proceed' to continue, or 'quit' to quit now -> proceed
        Updating firmware. This may take several minutes.
        Firmware update passed. Resetting HSM

Command Result : No Error
```

Figure 11-3: Example successful FW update console output from LunaCM

> **NOTE** All updates of the firmware MUST be FIPS validated before they can be loaded for the module to remain in an approved mode of operation.

11.12 Checking Module Status

The module status can be assessed by running either the `hsm showinfo` LunaCM for Thales Luna PCIe HSM or the `hsm show` LunaSH command for Thales Luna Network HSM.

Where physical access to the module is possible – additional information on status can be found based on the four status LED on the PCIe card as covered in section 3.1, 'Ports and Interface Overview'.

11.13 Maintenance Requirements

The module does not require any periodic maintenance outside the routine inspection of Tamper Evidence as documented in Section 7.2.

12 Mitigation of Other Attacks

No assured mitigations to ‘other attacks’ are covered in this security policy.

13 Guidance

13.1 Identifying the Module Version

Ahead of putting the module into its approved mode of operation, it is important to identify the hardware, main firmware and bootloader versions of the target module and to check these correspond to one of the tested modules listed in section Table 2-1. The following sections provide guidance on checking each element.

> **NOTE** Any module returning hardware, main firmware and bootloader versions not listed in this security policy is out of the scope of this validation and requires a separate FIPS 140-3 certificate.

Two paths are supported to checking the hardware, bootloader and main firmware versions depending on whether the LunaCM or LunaSH management interfaces are being used[50]:

- `hsm showinfo` when using LunaCM; and
- `hsm show` when using LunaSH.

Both commands return status information on the target cryptographic module including the module name, version numbers for both bootloader and main firmware and separately the hardware identity. Example output for each command for a valid module is shown in the Figure 13-1 and Figure 13-2 below with relevant versions highlighted in red:

```
lunacm:>hsm showinfo

        Partition Label -> myPCIeHSM
        Partition Manufacturer -> SafeNet
        Partition Model -> Luna K7
        Partition Serial Number -> 67842
        Partition Status -> L3 Device
        HSM Part Number -> 808-000048-002
        Token Flags ->
                CKF_RNG
                CKF_RESTORE_KEY_NOT_NEEDED
                CKF_TOKEN_INITIALIZED
        RPV Initialized -> Not Supported
        Slot Id -> 4
        Session State -> CKS_RW_PUBLIC_SESSION
        Role Status ->  none logged in
        Token Flags ->
                TOKEN_KCV_CREATED
        Partition OUID: 000000000000000002090100

        Partition Storage:
                Total Storage Space:  393216
                Used Storage Space:   0
                Free Storage Space:   393216
                Object Count:         0
                Overhead:             9848

        FM HW Status ->         FM
        Bootloader Version -> 1.1.5
        Firmware Version -> 7.8.5
        Rollback Firmware Version -> 7.0.3
```

[50] Both LunaCM and LunaSH map to the Luna ICD logical interface at the cryptographic module boundary.

```
        Environmental:
                Fan 1 Status                            : active
                Fan 2 Status                            : active
                Battery Voltage                         : 3.093 V
                Battery Warning Threshold Voltage       : 2.750 V
                System Temp                             : 38 deg. C
                System Temperature Warning Threshold    : 75 deg. C

        HSM Storage:
                Total Storage Space:  33554432
                Used Storage Space:   33554432
                Free Storage Space:   0
                Allowed Partitions:   1
                Number of Partitions: 1

        License Count -> 8
                1. 621000068-000 K7 Base Configuration
                2. 621010185-003 Key backup via cloning protocol
                3. 621000135-002 Enable allow decommissioning
                4. 621000134-002 Enable 32 megabytes of object storage
                5. 621000154-001 Enable decommission on tamper with policy off
                6. 621000021-002 Maximum performance
                7. 621000138-001 Controlled tamper recovery
                8. 621000074-001 Enable Functionality Modules

Command Result: No Error
```

Figure 13-1: Example output of `hsm showinfo` command from LunaCM

```
lunash:>hsm show

   Appliance Details:
   ==================
   Software Version:                7.8.0-1

   HSM Details:
   ============
   HSM Label:                          myLunaHSM
   Serial #:                           66331
   Bootloader:                         1.1.5
   Firmware:                           7.8.5
   HSM Model:                          Luna K7
   HSM Part Number:                    808-000073-001
   Authentication Method:              Password
   HSM Admin login status:             Not Logged In
   HSM Admin login attempts left:      3 before HSM zeroization!
   RPV Initialized:                    No
   Audit Role Initialized:             No
   Remote Login Initialized:           No
   Manually Zeroized:                  No
   Secure Transport Mode:              No
   HSM Tamper State:                   No tamper(s)

   Partitions created on HSM:
   ==============================
   Partition:         154438865296, Name: mypar0

   Number of partitions allowed:        100
   Number of partitions created:        1

   FIPS 140 Operation:
   =====================
   The HSM is in FIPS 140 approved operation mode.
   HSM Storage Information:
   ========================
   Maximum HSM Storage Space (Bytes):    33554432
   Space In Use (Bytes):                 335544
   Free Space Left (Bytes):              33218888


   Environmental Information on HSM:
   =================================
   Battery Voltage:                      3.072 V
   Battery Warning Threshold Voltage:    2.750 V
   System Temp:                          53 deg. C
   System Temp Warning Threshold:        75 deg. C


   Functionality Module HW:             FM
   =======================

Command Result: 0 (Success)
```

Figure 13-2: Example output of `hsm show` **command from LunaSH**

Where `Luna K7` is returned as the `Partition Model` in the output from LunaCM and `HSM Model` in the output from LunaSH, this can be mapped directly as an alias for ‘Thales Luna K7 Cryptographic Module’ as the module name.

13.2 Approved Mode of Operation

The module is configured to be in an Approved Mode of Operation on a per-partition basis.

To place a partition into its approved mode of operation, the HSM SO (Admin Partition) or Partition SO (User Partition) must check and, if necessary, set the following partition level policy:

> **Partition Policy (42) Enable CPv1**– this is set to `true` by default when the **HSM Policy (12), Allow Non FIPS Algorithms** is set to `true` or is set to `false` (enforced by the module) if the same policy is set to `false`. The policy shall be set to `false`.

> **Partition Policy (43) Enable non-FIPS Algorithms -** this policy is set to `true` by default if HSM Policy (12), Allow Non-FIPS Algorithms is separately set to true. If **HSM Policy (12), Allow Non-FIPS Algorithms** is set to `false`, the module will set this value to `false` (enforced by the module). This policy shall be set to `false`.

Ahead of configuring the individual partitions, the HSM SO must set the following HSM level policies:

> **HSM Policy (56), Allow User Defined ECC Curves** – this policy is set to `true` by default and shall be set to `false`.

NOTE If the HSM SO or Partition SO attempt to enable or disable the above policies, a warning is displayed and the HSM SO or Partition SO is prompted to confirm the selection. If this policy is left set to `true`, the module will be operating in the non-approved mode of operation.

> **HSM Policy (52) Restrict FM Privilege -** must be set to `true` following load of the ‘Enable Functionality Module CUF’ in order to ensure all FIPS 140-3 requirements are enforced at the boundary between the module firmware and any loaded functionality module.

NOTE While default setting is `false,` this setting is only relevant to a FIPS approved configuration following load of the ‘Enable Functionality Module CUF’. This unlocks the ability to enable the related ‘**HSM Policy (50) Allow Functionality Modules**’. Prior to HSM Policy (50) being set to `true`, the setting of **HSM Policy (52) Restrict FM Privilege** is redundant to the modules operation as FM can’t be loaded and the policy only relates to commands received by the module from within code executing within a sandboxed functionality module.

Following entry into an approved mode of operation:

- any changes to **the HSM level** policy will trigger an automatic zeroization of the HSM erasing all roles and partition stored key objects; and
- any changes to the partition level policy will trigger an automatic zeroization of the partition erasing all partition stored key objects.

13.3Using CA_PerformSelftest

To make the module perform cryptographic self-tests ‘on demand’ this can be achieved using the following:

- `Self Test` (Option #90) when using the `ckdemo` client tool.

This will give you 3 options of self-tests to run on the module:

- Option 1, H/W Test – runs the hardware self-tests.
- Option 2, Crypto Test – runs the cryptographic self-tests.
- Option 3, RNG Test – runs the statistical self-tests.
- All other options shown are not functional and return an error.

```
Enter your choice : 90
Slots available:
    slot#5 - Admin Token Slot
    slot#6 - User Token Slot
    slot#9 - Admin Token Slot
Select a slot (last selected slot = 6): 6
Test to perform:
[0] To cancel
[1] H/W Test
[2] Crypto Test
[3] RNG Test
[4] Perf mode on (us)
[5] Perf mode on (ns)
[6] Perf mode off
[7] Cryptographic Algorithm Self Tests
[8] Sentry off
[9] Sentry on
[10] Inject error: exit()
[11] Inject error: raise()
[12] Inject error: kernel oops
```

```
[13] Inject error: infinite loop
[14] List all enabled Sentry PKA engines (0:5)
[15] Disable a Sentry PKA engine (0:5)
[16] Enable a Sentry PKA engine (0:5)
> 2
Status: Doing great, no errors (CKR_OK)
(TITLE) menu titles, (99 or FULL) Full Help, (NONE) No help, (0 or EXIT) Quit
Status: Doing great, no errors (CKR_OK)
```

13-3: Example output of the `Self Test` command from CKDemo

In addition to using CKDemo as a tool, on demand self-test can be requested using the `CA_PerformSelfTest` command using the Cryptoki API.

13.4 Nominal Ranges

The nominal temperature range for operation of the module is between 0 and 50°C.

The nominal temperature range for storage and distribution of the module is between -20 and 80°C.

The overall input operation voltage of the module as a PCI-E card is +12V ± 8% DC. This input is used to generate the voltages covered in section 7.3 and as are used internal to the module.

The module’s tamper thresholds can be found in section 7.3.

13.5 Assuming Roles

To log into the module’s roles as described in section 4.1 you must first set the slot to the partition you wish to log into. To set the slot you must first use the `slot list` LunaCM command to view the available slots, once you have the chosen slot you wish to log into use the following:

> `set slot` when using LunaCM.

```
lunacm:> slot set -slot 4
Command Result : No Error
```

13-4: Example output of `set slot` command from LunaCM

Once set you may now log into any of the available roles to the partition by using:

> `role login` when using LunaCM.

```
lunacm:> role list
        Roles                   (short)
        ============================
        Partition SO              po
        Crypto Officer            co
        Limited Crypto Officer    lco
        Crypto User               cu

Command Result : No Error
```

```
lunacm:>role login -name po

        Please attend to the PED.

Command Result : No Error
```

13-5: Example output of `role list` and `role login` commands from LunaCM

> **NOTE** Some roles must first be initialized before they can be logged in. To do so you must use the `role init` command in LunaCM

13.6 Additional Guidance

In addition to the direct guidance provided in this security policy both Thales Luna PCIe HSM and Thales Luna Network HSM include extensive user guidance in their online free to access manual.

The full manuals for these products can be accessed at www.thalesdocs.com where the target products can be found under ‘Luna HSMs’ and where the ‘read docs’ link will take you to the front page where the document portal for either Thales Luna PCIe HSM and Thales Luna Network HSM can be found.

As part of the product documentation:

- **HSM Administration Guide** – describes how to install your Luna PCIe HSM adapter in a host workstation, install the Luna HSM Client software, and configure the HSM for use with your cryptographic applications;
- **Partition Administration Guide** – describes how to install Luna HSM Client and configure the application partition on the HSM to create and store your cryptographic objects and perform cryptographic operations;
- **LunaCM Command Reference** – describes how to access and use the LunaCM command line tool and provides detailed syntax descriptions for each available command; **LunaSH Command Reference** – describes how to access and use the LunaSH command line tool and provides detailed syntax descriptions for each available command. This interface is only available on Thales Luna Network HSM;
- **RestAPI References** – provides instructions and examples for setting up the REST API that can be used as an alternative interface to LunaSH;
- **SDK Reference** – describes how to use the Luna HSM SDK to integrate your applications with a Luna HSM; and
- **FM SDK Reference** – describes how to write, test, install, and use functionality modules to provide custom functions on a Luna HSM.

> **NOTE** When reviewing the manuals, you should refer to the latest version of each manual.
>
> Should any conflict be identified between guidance in this security policy and statements in the online product documentation, guidance in the security policy takes precedence.