THALES
Building a future we can all trust

Thales Luna G7 Cryptographic Module

LEVEL 3 NON-PROPRIETARY SECURITY POLICY

002-000179-001
Rev. K
January 10, 2025

Document Information

Document Part Number	002-000149-002
Release Date	January 10, 2025
Revision Version	K

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

Term	Definition
AES	Advanced Encryption Standard
AEK	AccessID Encryption Key
AEK-KW	AEK-Key Wrapping
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
CPV3	Cloning Protocol Version 3
CSP	Critical Security Parameter
CTR	CounTeR
CU	Crypto User
CVL	Component Validation List
DEK	Data Encryption Key
DH	Diffie-Hellman
DMK	Data MAC Key
DPK	Data Protection Key

Term	Definition
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
EMI	ElectroMagnetic Interference
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode
GMAC	Galois Message Authentication Code
HMAC	Keyed-Hash Message Authentication Code
HA	High Availability
HOC	Hardware Origin Certificate
HOK	Hardware Origin Key
HSM	Hardware Security Module / Host Security Module
ICD	Interface Control Design/Document
IG	Implementation Guidance
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
IV	Initialization Vector
KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-Based Key Derivation Function

Term	Definition
KCV	Key Cloning Vector
KDF	Key Derivation Function
KDM	Key Destruction Method
KEV	Key Encryption Vector
KTS	Key Transport Scheme
KW	Key Wrap
KWP	Key Wrap with Padding
LCO	Limited Crypto Officer
LED	Light Emitting Diode
MAC	Message Authentication Code
MGF	Mask Generation Function
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

Term	Definition
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
RSASVE	RSA Secret-Value Encapsulation
SHA	Secure Hash Algorithm
SKA	Static Key Agreement
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
TWC	Token or Module Wrapping Certificate
USB	Universal Serial Bus
USK	User’s Storage Key
XTS	XEX Tweakable block cipher ciphertext Stealing

REFERENCES

[FIPS 140-3] Federal Information Processing Standards Publication 140-3, Security Requirements for Cryptographic Modules, March 2019.

[FIPS 140-3 IG] NIST, Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program, January 29, 2024.

[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 186-5] Federal Information Processing Standards Publication 186-5, Digital Signature Standards (DSS), NIST, February 2023.

[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.

[SP800-56Cr1] NIST Special Publication 800-56C, Recommendation for Key-Derivation Methods in Key-Establishment Schemes, Revision 1, April 2018.

[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 (Revised), 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.

[SP800-133r2] 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.

[SP800-140Cr2] NIST Special Publication 800-140C, Revision 2, CMVP Approved Security Functions: CMVP Validation Authority Updates to ISO/IEC 24759, July 2023.

[SP800-140Dr2] NIST Special Publication 800-140D, Revision 2, CMVP Approved Sensitive Security Parameter Generation and Establishment Methods: CMVP Validation Authority Updates to ISO/IEC 24759, July 2023.

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

[ISO 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 G7 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 G7 Cryptographic Module meets all level 3 security requirements for [FIPS 140-3] as summarized in the table below:

Table 1-1: Security Levels

[ISO 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 G7 Cryptographic Module is a standalone hardware security module in the form of a USB device. The cryptographic module is contained in its own secure enclosure, which provides physical resistance.

The cryptographic boundary of the module is defined to encompass all components inside the secure enclosure in the USB device.

The module must be explicitly configured to operate in an Approved Mode of Operation using steps outlined in sections 13.3 and 13.4. Configuration steps outlined in these sections 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 G7 Cryptographic Module does not support degraded operation as defined in [ISO 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 19790:2012] is a **hardware module** with a **multi-chip standalone** embodiment.

NOTE The Thales Luna G7 Cryptographic Module can be used as follows:

- as a standalone device called the Thales Luna G7 USB HSM; or
- as a standalone device called the Thales Luna G7 Backup HSM.

The Tested Operational Environment's Physical Perimeter (TOEPP) of the modules is shown in Figure 2-1 and Figure 2-2 below. The TOEPP is defined as the enclosure on the top and bottom sides of the USB device defined as outlined.

Figure 2-1: Thales Luna G7 Cryptographic Module HW 808-000064-005 and 808-000064-006

Figure 2-2: Thales Luna G7 Cryptographic Module HW 808-000080-001 and 808-000080-002

The module includes a CR2032, 3V, coin battery, which is excluded from the TOEPP boundary. This battery is used to power the modules real-time clock when not connected through the 5V power interface. The real-time clock is not used to support any approved security functions supported by the module.

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

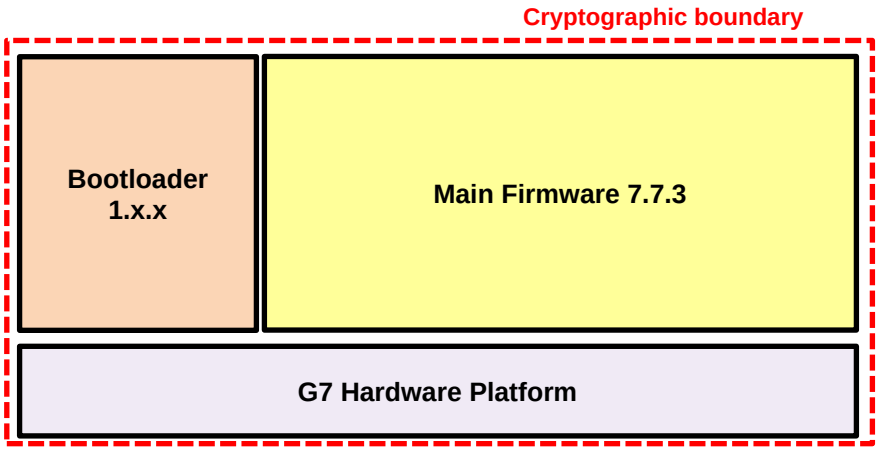

Figure 2-3: Thales Luna G7 Cryptographic Module cryptographic boundary.

The cryptographic boundary includes the bootloader and the main firmware. The bootloader and main firmware are 'firmware' within the scope of definitions in [ISO 19790:2012].

> **NOTE** As covered above, 'firmware' for the module includes the bootloader and main firmware. To use the module in an approved mode of operation, all firmware, including the bootloader and main firmware must be validated to [FIPS 140-3] to run on Thales Luna G7 Cryptographic Module.

2.3 Test Configuration

The following tested configurations are covered in this security policy:

Table 2-1: Cryptographic module configuration.

Model	Hardware [Part Number and Version]	Firmware Version	Distinguishing Features
Thales Luna G7 USB HSM or Thales Luna G7 Backup HSM	808-000080-001	Main firmware: 7.7.3 with Bootloader: 1.3.0, 1.5.0 or 1.6.0	Second USB port and screen with higher resolution.
Thales Luna G7 USB HSM or Thales Luna G7 Backup HSM	808-000080-002	Main firmware: 7.7.3 with Bootloader: 1.3.0, 1.5.0 or 1.6.0	Second USB port and screen with higher resolution. Functionally equivalent to 808-000080-001 with the difference limited to the supply choice for one of the non-security enforcing internal components.
Thales Luna G7 Backup HSM	808-000064-005	Main firmware: 7.7.3 with Bootloader: 1.3.0, 1.5.0 or 1.6.0	Single USB port and lower resolution screen.
Thales Luna G7 Backup HSM	808-000064-006	Main firmware: 7.7.3 with Bootloader: 1.3.0, 1.5.0 or 1.6.0	Single USB port and lower resolution screen. Functionally equivalent to 808-000064-005 with the difference limited to the supply choice for one of the non-security enforcing internal components.

This document covers both the PED and password authentication configurations of the Thales Luna G7 Cryptographic Module.

NOTE The security features described in this document apply to the Thales Luna G7 Cryptographic Module only and do not include any feature that may be enforced by the client or Thales Luna PED.

NOTE As the module is a hardware module of 'multi-chip standalone' embodiment, 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 #C2022) and (Cert #A6549); and
- **SafeNet Cryptographic Library** (Certs #C2020, #A674 and #A2125).

The following entropy noise source and associated ESV certificate is used by the cryptographic module:

- **Thales G7 Hardware Platform TRNG** (Cert #E97).

The approved algorithms implemented by the module, alongside their mapping to the certificates above, and together with 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.

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

- Cert #C2020 includes listing for KAS-ECC, KAS-FFC, CVL (RSADP) and GMAC;;
- Cert #C2022 includes SHA1 and SHA2-256. The implementations for SHA1 and SHA2-256 are present in the bootloader to verify firmware integrity but are redundant to other approved security functions. Firmware authenticity is checked using RSA PKCS#1 v1.5 with SHA2-384.

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 these functions are either not called by the modules executable code or the target function is not used as a security function to satisfy requirements from FIPS 140-3.

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A2125	ECDSA SigGen **Standard:** [FIPS 186-4]	Signature Generation SHA3-224, SHA3-256, SHA3-384, SHA3-512.	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, Generate signature or MAC over user supplied data , Setup Remote PED Session.
#A2125	ECDSA SigVer **Standard:** [FIPS 186-4]	Signature Verification SHA3-224, SHA3-256, SHA3-384, SHA3-512.	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, Verify signature or MAC over user supplied data, Setup Remote PED Session.
#A2125	HASH_DRBG **Standard:** [SP800-90Ar1].	SHA2-256	256-bits.	Clone SMK between partitions, Configure partition for high-available recovery / login, Create a user partition, Enable/disable STM, Export secret or private key using key wrapping, Extract entropy from DRBG, Generate domain parameters, Generate local symmetric or asymmetric key-pair, Generate signature or MAC over user supplied data, Initialize the HSM, Initialize Remote PED Vector (RPV), Initialize role, Perform encrypt operation on user supplied data object, Request HSM self-test, Setup Remote PED Session, Re-seed partition DRBG, Rollover SMK for a given partition.
#A2125	KAS-ECC **Standard:** [SP800-56Ar3]	fullUnified with full key validation and key-pair generation **KDF:** OneStep using SHA2-512. **Key Confirmation:** HMAC-SHA2-256 with 256-bit key.	P-521.	Setup Remote PED Session.
#A2125	KAS-ECC-SSC **Standard:** [SP800-56Ar3].	ephemeralUnified, fullUnified, onePassDH	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.
#A2125	KAS-FFC-SSC **Standard:** [SP800-56Ar3]	dhHybrid1, dhEphem, dhHybridOneFlow, dhOneFlow.	2048, 3072 and 4096-bits.	Request HSM self-test, Derive key from existing partition secret or private key object.
#A2125	KAS-IFC **Standards:** [SP800-56Br2] and [SP800-56Cr2].	KAS1-basic **Key generation method:** rsakpg1-crt, rsakpg2-crt. **KDF method**: One-Step Key Derivation from [SP800-56Cr2] using SHA2-512.	4096-bits.	Clone SMK between partitions, Configure partition for high-available recovery / login.

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A2125	KDA One-Step Sp800-56Cr1 **Standard:** [SP800-56Cr1].	One-Step Key Derivation SHA1.	**Shared secret length:** 224-8192, increment 1 byte. **Derived Key length:** 512.	Request HSM self-test, Derive key from existing partition secret or private key object.
#A2125	KDA One-Step Sp800-56Cr2 **Standard:** SP800-56Cr2.	One-Step Key Derivation 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:** 4096-bits.	Request HSM self-test, Derive key from existing partition secret or private key object, Clone SMK between partitions.
#A2125	KDF ANS 9.42 (CVL)[1] **Standard:** [SP800-135r1]**.**	SHA1, 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.	Request HSM self-test, Derive key from existing partition secret or private key object.
#A2125	KDF ANS 9.63 (CVL) **Standard:** [SP800-135r1]**.**	SHA2-224, SHA2-256, SHA2-384, SHA2-512.	128, 4096 bits.	Perform encrypt operation on user supplied data object.
#A2125	KTS-IFC **Standards:** [SP800-56Br2] and [SP800-56Cr2].	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	2048, 3072, 4096, 6144, and 8192 bits. **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, Initialize the HSM, Initialize role, Change authentication data, Login as role.
#A2125	PBKDF[2] **Standard:** [SP800-132].	HMAC-SHA2-512	**Derived Key Length:** 256-bits **Password Length:** 128-bits **Salt Length:** 256-bits	Initialize the HSM, Initialize Remote PED Vector (RPV), Initialize role, Change authentication data, Login as role, Request HSM self-test.
#A6549	RSA SigVer **Standard:**. [FIPS 186-5]	Signature Verification SHA2-384	4096-bit	Request authentication and execution of main firmware.

[2] 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). Further information on use of PBKDF is provided in section 4.2.1.

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A674	RSA KeyGen **Standard:** [FIPS 186-4].	Key Generation B.3.3 and B.3.6	4096 bits. **Vendor Note**: Key sizes up to modulus length 8192-bit are supported for key generation 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.	Configure partition for high-available recovery / login, Generate local symmetric or asymmetric key-pair, Initialize the HSM, Initialize role, Request HSM self-test.
#A674	RSA SigGen **Standard:** [FIPS 186-4].	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. (ANSI X9.31): 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.**	4096-bit. **Vendor Note**: Key sizes up to modulus length 8192-bit are supported for key generation 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.	Clone SMK between partitions, Generate signature or MAC over user supplied data, Request HSM self-test.
#A674	RSA SigVer **Standard:** [FIPS 186-4].	Signature Verification (PKCS #1-v1.5 and PKCS-PSS): SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512 (ANSI X9.31): SHA1, 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.**	4096-bit. **Vendor Note:** Key sizes 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.	Load configuration update file, Clone SMK between partitions, Configure partition for high-available recovery / login, Generate local symmetric or asymmetric key-pair, Request HSM self-test, Update firmware, Verify signature or MAC over user supplied data.
#C2020	AES-CBC **Standards:** [FIPS 197].	CBC	128, 192, 256 bits.	Perform encrypt operation on user supplied data object, Perform decrypt operation on user supplied data object, Import secret or private key using key wrapping, Insert key from external storage using SKS, , Request HSM self-test.

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C2020	AES-CFB128 **Standards:** [FIPS 197] and [SP800-38A].	CFB128	128, 192, 256 bits.	Perform encrypt operation on user supplied data object, Perform decrypt operation on user supplied data object, Request HSM self-test.
#C2020	AES- CFB8 **Standards:** [FIPS 197] and [SP800-38A].	CFB8	128, 192, 256 bits.	Perform encrypt operation on user supplied data object, Perform decrypt operation on user supplied data object, Request HSM self-test.
#C2020	AES-CMAC **Standards:** [FIPS 197], [SP800-38D], [SP800-38E] and [SP800-38F].	CMAC	128, 192, 256 bits.	Generate signature or MAC over user supplied data, Verify signature or MAC over user supplied data, Request HSM self-test.
#C2020	AES-CTR **Standards:** [FIPS 197] and [SP800-38A].	CTR	128, 192, 256 bits.	Send or receive data over PED tunnel (remote PED), Perform encrypt operation on user supplied data object, Perform decrypt operation on user supplied data object¸ Import secret or private key using key wrapping, Request HSM self-test.
#C2020	AES-ECB **Standards:** [FIPS 197] and [SP800-38A].	ECB	128, 192, 256 bits.	Perform encrypt operation on user supplied data object, Perform decrypt operation on user supplied data object, Import secret or private key using key wrapping, Request HSM self-test.
#C2020	AES-GCM[3] **Standards:** [FIPS 197] and [SP800-38D].	GCM	128, 192, 256 bits.	Extract key to external storage using SKS, Perform encrypt operation on user supplied data object, Perform decrypt operation on user supplied data object, Import secret or private key using key wrapping, Export secret or private key using key wrapping, Request HSM self-test.
#C2020	AES-KW **Standards:** [FIPS 197] and [SP800-38F].	KW	128, 192, 256 bits.	Perform encrypt operation on user supplied data object, Perform decrypt operation on user supplied data object, Import secret or private key using key wrapping, Export secret or private key using key wrapping, Request HSM self-test.

[3] The module generates IVs internally using the approved DRBG where all IV used are 128-bits in length per [SP800-38D] and in accordance with FIPS 140-3 I.G. C.H, scenario 2.

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C2020	AES-KWP **Standards:** [FIPS 197] and [SP800-38F].	KWP	128, 192, 256 bits.	Initialize the HSM, Create a user partition, Initialize role, Export/import audit log secret key, Clone SMK between partitions, Rollover SMK for a given partition, Change authentication data, Initialize role, Configure partition for high-available recovery / login, Login as role, Initialize Remote PED Vector (RPV), 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, Verify signature or MAC over user supplied data, Change authentication data , Request HSM self-test.
#C2020	AES-OFB **Standards:** [FIPS 197] and [SP800-38A].	OFB	128, 256 bits.	Perform encrypt operation on user supplied data object, Perform decrypt operation on user supplied data object, Request HSM self-test.
#C2020	DSA KeyGen **Standard:** [FIPS 186-4].	Key Generation	2048, 3072 bits.	Generate local symmetric or asymmetric key-pair.
#C2020	DSA PQGGen **Standard:** [FIPS 186-4].	Parameter Generation: SHA2-224, SHA2-256	2048, 3072 bits.	Generate domain parameters.
#C2020	DSA SigGen **Standard:** [FIPS 186-4].	Signature Generation SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	2048, 3072 bits.	Generate signature or MAC over user supplied data, Request HSM self-test.
#C2020	DSA SigVer **Standard:** [FIPS 186-4].	Signature Verification SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	1024, 2048, 3072 bits.	Verify signature or MAC over user supplied data, Request HSM self-test.

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C2020	ECDSA KeyGen **Standard:** [FIPS 186-4].	Key Generation	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521.	Generate local symmetric or asymmetric key-pair, Initialize Remote PED Vector (RPV), Setup Remote PED Session.
#C2020	ECDSA SigGen **Standard:** [FIPS 186-4].	Signature Generation SHA2-224, SHA2-256, SHA2-384, SHA2-512.	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521.	Generate signature or MAC over user supplied data, Request HSM self-test, Setup Remote PED Session,.
#C2020	ECDSA SigVer **Standard:** [FIPS 186-4].	Signature Verification SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512.	B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521.	Request HSM self-test, Setup Remote PED Session, Verify signature or MAC over user supplied data.
#C2020	HMAC-SHA-1 **Standards:** [FIPS 198-1] and [FIPS 202].	HMAC-SHA-1	**Mac size:** 80, 96, 128, 160. **Key size:** key size < block size, key size = block size, key size > block size.	Generate signature or MAC over user supplied data, Request HSM self-test, Verify signature or MAC over user supplied data.
#C2020	HMAC-SHA2-224 **Standards:** [FIPS 198-1] and [FIPS 180-4].	HMAC-SHA2-224	**Mac size:** 112, 128, 160, 192, 224. **Key size:** key size < block size, key size = block size, key size > block size.	Generate signature or MAC over user supplied data, Verify signature or MAC over user supplied data, Request HSM self-test.
#C2020	HMAC-SHA2-256 **Standards:** [FIPS 198-1] and [FIPS 180-4].	HMAC-SHA2-256	**Mac size:** 128, 192, 256. **Key size:** key size < block size, key size = block size, key size > block size.	Send or receive data over PED tunnel (remote PED), Request HSM self-test, Generate signature or MAC over user supplied data, Verify signature or MAC over user supplied data, Generate secure log record.
#C2020	HMAC-SHA2-384 **Standards:** [FIPS 198-1] and [FIPS 180-4].	HMAC-SHA2-384	**Mac size:** 192, 256, 320, 384. **Key size:** key size < block size, key size = block size, key size > block size.	Generate signature or MAC over user supplied data, Request HSM self-test, Verify signature or MAC over user supplied data.
#C2020	HMAC-SHA2-512 **Standards:** [FIPS 198-1] and [FIPS 180-4].	HMAC-SHA2-512	**Mac size:** 256, 320, 384, 448, 512. **Key size:** key size < block size, key size = block size, key size > block size.	Generate signature or MAC over user supplied data, Request HSM self-test, Verify signature or MAC over user supplied data.
#C2020	HMAC-SHA3-224 **Standards:** [FIPS 198-1] and [FIPS 202].	HMAC-SHA3-224	**Mac size:** 112, 128, 160, 192, 224. **Key size:** key size < block size, key size = block size, key size > block size.	Generate signature or MAC over user supplied data, Request HSM self-test, Verify signature or MAC over user supplied data.
#C2020	HMAC-SHA3-256 **Standards:** [FIPS 198-1] and [FIPS 202].	HMAC-SHA3-256	**Mac size:** 128, 192, 256. **Key size:** key size < block size, key size = block size, key size > block size.	Generate signature or MAC over user supplied data, Request HSM self-test Verify signature or MAC over user supplied data.
#C2020	HMAC-SHA3-384 **Standards:** [FIPS 198-1] and [FIPS 202].	HMAC-SHA3-384	**Mac size:** 192, 256, 320, 384. **Key size:** key size < block size, key size = block size, key size > block size.	Generate signature or MAC over user supplied data, Request HSM self-test Verify signature or MAC over user supplied data.

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C2020	HMAC-SHA3-512 **Standards:** [FIPS 198-1] and [FIPS 202].	HMAC-SHA3-512	**Mac size:** 256, 320, 384, 448, 512. **Key size:** key size < block size, key size = block size, key size > block size.	Generate signature or MAC over user supplied data, Request HSM self-test Verify signature or MAC over user supplied data.
#C2020	KDF **Standard:** [SP800-108r1].	KBKDF **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	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.
#C2020	RSA KeyGen **Standard:** [FIPS 186-4]	Key Generation B.3.3 and B.3.6	2048 and 3072 bits.	Request HSM self-test, Generate local symmetric or asymmetric key-pair.
#C2020	RSA SigGen **Standard:** [FIPS 186-4].	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 (ANSI X9.31): 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.**	2048 and 3072 bits.	Request HSM self-test, Generate signature or MAC over user supplied data.

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C2020	RSA SigVer **Standard:** [FIPS 186-4].	Signature Verification (PKCS #1-v1.5 and PKCS-PSS): SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. (ANSI X9.31): SHA1, 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.**	1024, 2048 and 3072 bits.	Request HSM self-test, Verify signature or MAC over user supplied data. Legacy[4] for 1024 bits
#C2020	SHA1 **Standard:** [FIPS 180-4] and [FIPS 202].	SHA1	N/A	Request HSM self-test, Import secret or private key using key wrapping, Perform digest operation on user supplied data.
#C2020	SHA2-224 **Standard:** [FIPS 180-4] and [FIPS 202].	SHA2-224	N/A	Request HSM self-test, Import secret or private key using key wrapping, Perform digest operation on user supplied data.
#C2020	SHA2-256 **Standards:** [FIPS 180-4] and [FIPS 202].	SHA2-256	N/A	Request HSM self-test, Protect object integrity, Enable/disable STM, Initialize role, Configure partition for high-available recovery / login, Login as role, Import secret or private key using key wrapping, Insert key from external storage using SKS, Perform digest operation on user supplied data.
#C2020	SHA2-384 **Standard:** [FIPS 180-4].	SHA2-384	N/A	Request HSM self-test, Update firmware, Load configuration update file, Configure partition for high-available recovery / login, Import secret or private key using key wrapping, Perform digest operation on user supplied data.

[4] Legacy usage only. These legacy algorithms can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M.

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C2020	SHA2-512 **Standard:** [FIPS 180-4].	SHA2-512	N/A	Request HSM self-test, Clone SMK between partitions, Enable/disable STM, Initialize role, Configure partition for high-available recovery / login, 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, Perform digest operation on user supplied data, Perform encrypt operation on user supplied data object, Generate signature or MAC over user supplied data.
#C2020	SHA3-224 **Standard:** [FIPS 202].	SHA3-224	N/A	Request HSM self-test, Import secret or private key using key wrapping, Perform digest operation on user supplied data.
#C2020	SHA3-256 **Standard:** [FIPS 202].	SHA3-256	N/A	Request HSM self-test, Import secret or private key using key wrapping, Perform digest operation on user supplied data.
#C2020	SHA3-384 **Standard:** [FIPS 202].	SHA3-384	N/A	Request HSM self-test, Import secret or private key using key wrapping, Perform digest operation on user supplied data.
#C2020	SHA3-512 **Standard:** [FIPS 202].	SHA3-512	N/A	Request HSM self-test, Import secret or private key using key wrapping, Perform digest operation on user supplied data.
#C2020	SHAKE-128 **Standard:** [FIPS 202].	SHAKE-128	N/A	Request HSM self-test, Perform digest operation on user supplied data.
#C2020	SHAKE-256 **Standard:** [FIPS 202].	SHAKE-256	N/A	Request HSM self-test, Perform digest operation on user supplied data.
#C2020	Triple-DES-CBC **Standards:** [SP800-67r2] and [SP800-38A].	CBC	168-bits	Request HSM self-test, Perform decrypt operation on user supplied data object, Import secret or private key using key wrapping. Legacy[4] Decryption
#C2020	Triple-DES-CFB64 **Standards:** [SP800-67r2] and [SP800-38A].	CFB64	168-bits	Request HSM self-test, Perform decrypt operation on user supplied data object. Legacy[4] Decryption

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#C2020	Triple-DES-CFB8 **Standards:** [SP800-67r2] and [SP800-38A].	CFB8	168-bits	Request HSM self-test, Perform decrypt operation on user supplied data object. Legacy[4] Decryption
#C2020	Triple-DES-CMAC **Standards:** [SP800-67r2] and [SP800-38B].	CMAC (MAC verify only)	168-bits	Request HSM self-test. Verify signature or MAC over user supplied data. Legacy[4] Verification
#C2020	Triple-DES-CTR **Standards:**. [SP800-67r2] and [SP800-38A].	CTR	168-bits	Request HSM self-test, Perform decrypt operation on user supplied data object, Import secret or private key using key wrapping. Legacy[4] Decryption
#C2020	Triple-DES-ECB **Standards:**. [SP800-67r2] and [SP800-38A].	ECB	168-bits	Request HSM self-test, Perform decrypt operation on user supplied data object, Import secret or private key using key wrapping. Legacy[4] Decryption
#C2020	Triple-DES-OFB **Standards:**. [SP800-67r2] and [SP800-38A].	OFB	168-bits	Request HSM self-test, Perform decrypt operation on user supplied data object. Legacy[4] Decryption
#C2022	SHA2-384 **Standard:**. [FIPS 180-4]	SHA2-384 (Byte Only)	N/A	Request authentication and execution of main firmware.
#A2125	KAS (KAS-ECC-SSC (Cert #A2125) and CVL (Cert #A2125)) **Standards:** [SP800-56Ar3], and [SP800-135r1]**.**	ephemeralUnified, and onePassDH with X9.63 KDF from [SP800-135r1] using SHA2-224, SHA2-256, SHA2-384, SHA2-512.	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.
#A2125	KAS (KAS-ECC-SSC (Cert #A2125) and KDA (Cert #A2125)) **Standards:** [SP800-56Ar3] and [SP800-56Cr2].	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, SHA3-512.	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.

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

Table 2-2: Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A2125	KAS (KAS-FFC-SSC (Cert #A2125) and CVL (Cert #A2125)) **Standards:** [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, SHA3-512.	2048, 3072 and 4096 bits. 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.
#A2125	KAS (KAS-FFC-SSC (Cert #A2125) and KDA (Cert #A2125)) **Standards:** [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, SHA3-512.	2048, 3072 and 4096 bits. 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.
#E97	**Physical** [SP800-90B], [FIPS 180-4]	Live noise source	Full Entropy	Initialize the HSM, Create a user partition, Clone SMK 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 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 DRBG, Perform encrypt operation on user supplied data object, Generate signature or MAC over user supplied data.

Table 2-3: Vendor Affirmed Approved Algorithms

Algorithm	Caveat	Use / Function
CKG[6] [SP800-133r2]	Vendor Affirmed	Initialize the HSM, Create a user partition, Clone SMK 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.

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

Algorithm	Caveat	Use / Function
Key Agreement Scheme		
KAS-ECC-SSC (Cert #A2125)	ephemeralUnified, fullUnified, onePassDH When using Non-NIST curves from Table 2-6 and allowances from [FIPS 140-3 IG] C.A, Use of Non-approved elliptic curves.	Derive key from existing partition secret or private key object.
Key Transport		
KTS (AES Cert. #C2020)	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 #C2020 without use of an additional approved hash function.	Clone SMK between partitions, Import secret or private key using key wrapping. Legacy[4] Unwrapping
KTS (Triple-DES Cert #C2020)	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 #C2020 without use of an additional approved hash function.	Import secret or private key using key wrapping. Legacy[4] Unwrapping

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

[6] Symmetric keys and seed for asymmetric key generation are created based on the direct output of the module DRBG (#C2020) based on [SP800-133r2] using example 1 from section 4 and 6.1 where V is a string of binary zeroes and as such B = V. Asymmetric Key pairs are generated based on [SP800-133r2], section 5.1 and 5.2 and using methods from [FIPS 186-4]. The module supports derivation of keys from other keys as per [SP800-133r2], section 6.2.2, and supports the derivation of keys from passwords using methods in 6.2.3 and [SP800-132] for storage encryption. Keys can also be recovered from key components entered into the module using methods from [SP800-133r2], section 6.3 and both concatenation of components (option 1) or Exclusive-Oring (option 2).

Table 2-6: 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
Brainpool P512r1	Prime field – GF(p)	[RFC 5639].	256-bits	X	X	X
Brainpool P512t1	Prime field – GF(p)	[RFC 5639].	256-bits	X	X	X
Brainpool P-384r1	Prime field – GF(p)	[RFC 5639].	192-bits	X	X	X
Brainpool P-384t1	Prime field – GF(p)	[RFC 5639].	192-bits	X	X	X
Brainpool P320r1	Prime field – GF(p)	[RFC 5639].	160-bits	X	X	X
Brainpool P320t1	Prime field – GF(p)	[RFC 5639].	160-bits	X	X	X
secp256k1	Prime field – GF(p)	[SEC 2].	128-bits	X	X	-
Brainpool P-256r1	Prime field – GF(p)	[RFC 5639].	128-bits	X	X	X
Brainpool P-256t1	Prime field – GF(p)	[RFC 5639].	128-bits	X	X	X
Brainpool P-224r1	Prime field – GF(p)	[RFC 5639].	112-bits	X	X	X
Brainpool P-224t1	Prime field – GF(p)	[RFC 5639].	112-bits	X	X	X

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 13.3 and 13.4).

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 module service.

Table 2-7: 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	

Algorithm / Function	Use / Function
RSA (non-compliant with less than 112 bits of encryption strength)	
RSA X.509[7]	
SEED	
SM4	
Triple-DES (non-compliant for encrypt operations)	
XOR[8]	
Hashing	
HAS-160	Derive key from existing partition secret or private key object, Verify signature or MAC over user supplied data, Perform digest operation on user supplied data.
KECCAK	
MD2	
MD5	
RIPEMD-160	
SM3	
Message Authentication Code	
AES-MAC	Generate signature or MAC over user supplied data, Verify signature or MAC over user supplied data.
ARIA-CMAC	
ARIA-MAC	
CAST3-MAC	
CAST5-MAC	
COMP128	
DES-MAC	
HAS160-HMAC	
HMAC (non-compliant with less than 112 bits of encryption strength)	
MD5-HMAC	
MILENAGE	
RC2-MAC	
RC5-MAC	
RIPEMD160-HMAC	
SEED-CMAC	
SEED-MAC	
SM3-HMAC	
SSL3-MD5-MAC	
SSL3-SHA1-MAC	

[7] 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.
[8] 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)	
Triple-DES-MAC	
Triple-DES-x9.19-MAC	
TUAK	
Asymmetric	
DSA (non-compliant with less than 112 bits of encryption strength)	Generate signature or MAC over user supplied data, Verify 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[9]	Derive key from existing partition secret or private key object.
ARIA	
BIP32	
DES	
MD5	
SHA[10]	
SSL PRE-MASTER	
SSL3-MASTER	
SM3	
Triple-DES	
XOR[11]	
Key Agreement	
Diffie-Hellman (key agreement; key establishment methodology; non-compliant with less than 112 bits of encryption strength)	Derive key from existing partition secret or private key object.
ECC (non-compliant with less than 112 bits of encryption strength)	
Key Transport	
AES[12]	Import secret or private key using key wrapping, Export secret or private key using key wrapping.
ARIA	

[9] 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.
[10] 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].
[11] XOR is non-approved for key derivation when selected as a mechanism to combine supplied user data with an existing module stored key.
[12] 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.

Algorithm / Function	Use / Function
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[13]	
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	

[13] 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 figure identifies the physical interfaces to the cryptographic module:

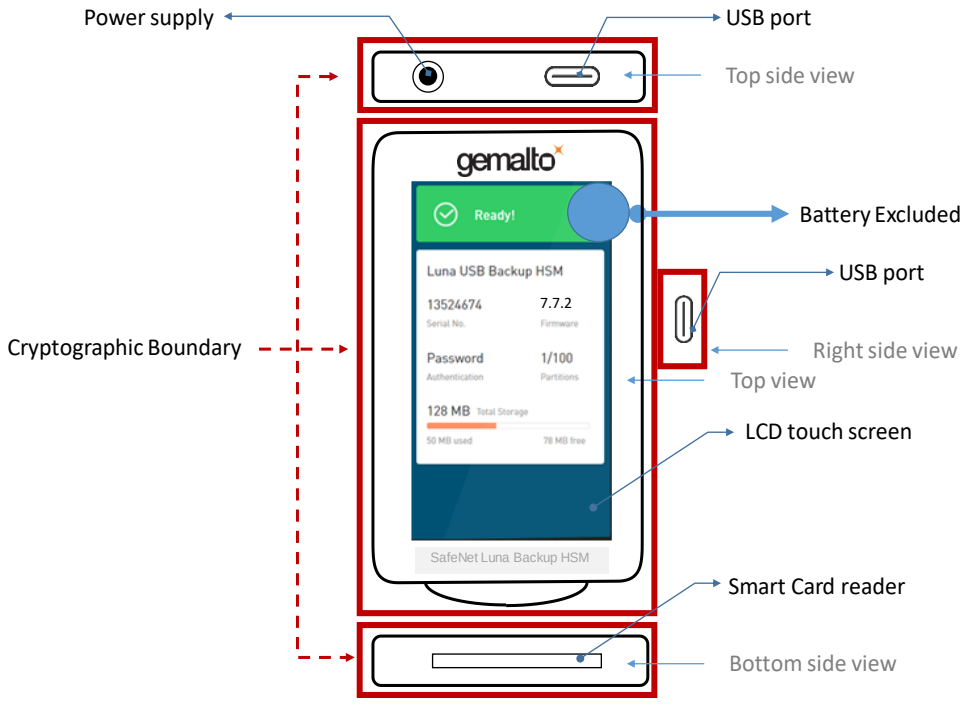

Figure 3-1: Thales Luna G7 Cryptographic Module physical interfaces.

> **NOTE** The USB port on the right side of the figure is only present on the 808-000080-001 and 808-000080-002 versions on the hardware.

The cryptographic module is a multi-chip standalone hardware module in the small form factor device. The cryptographic boundary of the module is shown above. The cryptographic boundary is defined to encompass all components inside the enclosure, including the LCD touch screen panel.

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
Top USB3 Type C port	Data input interface, data output interface, control input interface, status output interface	Diagnostics information when the 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
Right USB2 Type C port (Only present on 808-000080-001 and 808-000080-002 versions of the hardware)	Data input interface, data output interface, control input interface, status output interface	Channel for the iKey when connected locally
Smart Card reader	Data input interface, data output interface, control input interface, status output interface	This is redundant interface not currently used by the module. The interface is disabled in all current configurations of the module
LCD Touch Screen	Data input interface, data output interface, control input interface, status output interface	**During boot sequence:** > used to signal progress during the boot process **General operation:** > Display useful information to the user regarding the state of the device; > Primary interface for user interaction when the iKey is connected locally
Power supply 5V	Power interface	N/A

3.2 Trusted Channel

3.2.1 Trusted channel summary

Where CSP and authentication are not separately encrypted, these are exclusively passed into or output from the module by a trusted channel.

The following trusted channels are defined for Thales Luna G7 Cryptographic Module:

> **Physical Trusted Path (USB)** – the local USB interface is considered a physical trusted path into the module and is used to directly connect an iKey to the module.

> **Authentication Trusted Path (LCD)** – used for transport of authorization data from the user to the module.

All interfaces are only active for the input or output of CSPs in response to an active session over the ICD API.

The following sections describe each trusted path in more detail.

3.2.2 Physical Trusted Path (USB)

If configured, the module can use an iKey as an external data input/output device for CSP and authentication data. The iKey connects to the module's rightmost USB port and is used to pass authentication data and CSPs to and from the module via a physical trusted path. CSP's and authentication data that are output to the iKey are written to the iKey itself.

Authentication of the module by the iKey is done by physical means based on the direct physical connection of the iKey to the Thales Luna G7 Cryptographic Module. All authentication data from the user is entered via the module's LCD screen.

Messages exchanged from the iKey and the module are encrypted using AES with a 256-bit key in CTR mode alongside a MAC using HMAC-SHA2-256.

3.2.3 Authentication Trusted Path (LCD)

Authentication of the module is done using G7 UI via the liquid crystal display (LCD) of the Thales Luna G7 Cryptographic Module. This trusted channel is used as part of the identity based authorization scheme to present the authentication data and is not separately authenticated.

4 Roles, Services, and Authentication

4.1 Roles

The Thales Luna G7 Cryptographic Module supports the following roles:

Table 4-1: Thales Luna G7 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 and set and maintain global HSM security policies. 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 SO is responsible for selecting the authentication method during the HSM initialization. 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.

The act of logging into the roles can be found in section 13.7.

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 G7 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, Services, 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	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
Any role	Generate secure log record	session and app_ID, message to log, message type	return code

Role	Service	Input	Output
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
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
HSM SO, Partition CO	Clone SMK between partitions	session, SMK 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	Request HSM self-test	session, self-test ID	return code
Role Management			
Any role	Query role status	session, role	role status, return code
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

Role	Service	Input	Output
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 Remote PED Session	PED_ID	return code
Any role	Send or receive data over PED tunnel (remote PED)	when receiving data: plaintext payload when sending data: return code	when sending data: plaintext payload when receiving 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

Role	Service	Input	Output
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
HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU	Extract entropy from 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
HSM SO, Administrator, Partition CO, Partition LCO, Partition CU	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 CO, Partition LCO, Partition CU	Verify signature or MAC over user supplied data	session, algorithm, algorithm parameters, data to verify, signature	return code
Bootloader Services			
Any role	Request complete erase of the HSM main firmware image and key stores (excludes erase of bootloader)	-	-
Any role	Request authentication and execution of main firmware	-	-

4.2 Roles and Authentication

4.2.1 Authentication Mechanism Summary

All users, except for the Public User, must authenticate to the module using identity-based authentication.

If configured with PED, all roles must authenticate using an iKey. The iKey can be connected directly to the device or to the Luna PED device. In case of Luna PED, the connection needs to be done over the Remote PED channel. When a role is initialized, a module generates the authentication data as a 48-byte random value and writes it to an 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 Required Identification 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 are limited to a character set of 86 characters[14] presented to the module as there 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)[15].

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 8 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 8 characters is 1 in 2^{51}. This probability is significantly reduced if random passwords are not used.

[14] Supported characters in memorized secret are limited to:
`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 !@#$%^*()-_=+[]{}/:',.~`

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

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 an iKey. 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[16].

4.2.3 Account lockout behaviours

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 behaviours observed:

> lockout of the HSM SO role will trigger the HSM zeroize 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 **Table 2-6.**

When the module is operating in this mode, security functions in Table 2-7 are disabled and blocked from being used.

As notes on the content of **Table 4-4: Approved Services**:

> In the 'Approved Security Functions' column:

- 'Algorithms' maps the target service to cryptography from standards referenced in [SP800-140Cr2] alongside corresponding CAVP certificates from Table 2-2 or 'non-Approved but Allowed' cryptography from Table 2-6.

[16] A module is reset in response to a trigger signal being received on a request from a host application.

- ‘Key Management technique’ maps the target service to cryptography from standards referenced in [SP800-140Dr2] alongside corresponding CAVP certificates from Table 2-2 or ‘non-Approved but Allowed’ cryptography from Table 2-6.
- ‘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:
  - **For USB HSM** – Partition Policy (43), Allow non-FIPS Algorithms is set to `disabled` AND HSM Policy (56), Allow User Defined ECC Curves is set to `disabled` AND return code is `CKR_OK`; or
  - **For Backup HSM** – HSM Policy (55), Enable Restricted Restore is set to `enabled` AND return code is `CKR_OK`; and
- IND_2:
  - **For USB HSM** – Partition Policy (43), Allow non-FIPS Algorithms is set to `disabled` AND HSM Policy (56), Allow User Defined ECC Curves is set to `disabled` AND return code is `PED_RET_OK` or `SP_RET_OK`.; or
  - **For Backup HSM** – HSM Policy (55), Enable Restricted Restore is set to `enabled` AND return code is `PED_RET_OK` or `SP_RET_OK`.
- IND_3:
  - **For USB HSM** – Partition Policy (43) Allow non-FIPS Algorithms is set to `disabled` AND HSM Policy (56), Allow User Defined ECC Curves is set to `disabled` AND return code is `0`; or

  For Backup HSM – HSM Policy (55), Enable Restricted Restore is set to `enabled` AND return code is `0`.

> In the ‘Keys and/or SSPs’ column:

- For a complete description of SSP referenced from the table, see Table 9-1.

Table 4-4: Approved 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	CITS-DAC, CITS-DAK, ECC DAC, Secure Audit AccessID-HMAC Key, PSK, USK, DRBG C, DRBG V, Entropy Input String, Entropy Seed, , KCV, SMK, HA_{PUB}, HA_{PK}, RND, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), SALK, CWK_{HSM}, CWK_{PED}, DEK_{HSM}, DMK_{HSM}, DEK_{PED}, DMK_{PED}, , AEK, AEK-EK, AccessID.	Any role	**Z:** (for ALL partition) CITS-DAC, CITS-DAK, ECC DAC, Secure Audit AccessID-HMAC Key, PSK, USK, DRBG C, DRBG V, KCV, SMK. 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: SALK, CWK_{HSM}, CWK_{PED}, DEK_{HSM}, DMK_{HSM}, DEK_{PED}, DMK_{PED} **E:** AEK, AEK-EK, AccessID.	IND_1

Table 4-4: Approved Services

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 #C2020) – KWP mode with 256-bit key **Key management technique:** ESV (Cert #E97), CKG, HASH_DRBG (Cert #A2125), SHA #C2020) – SHA2-512, PBKDF (Cert #A2125), KBKDF (Cert #C2020) **Authentication technique:** N/A	For ALL partition if present – HOK, 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 C, DRBG V, Entropy Input String, Entropy Seed, , 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 C, DRBG V, Entropy Input String, Entropy Seed, KCV, Stored User Password Hash **G:** PED Authentication Data, User Password **E:** HOK, 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 #C2020) – KWP mode with 256-bit key **Key management technique:** ESV (Cert #E97), CKG, SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125) **Authentication technique:** N/A	DRBG C, DRBG V, Entropy Input String, Entropy Seed , KCV, AEK, AEK-EK, AccessID..	HSM SO	**G:** DRBG C, DRBG V, Entropy Input String, Entropy Seed **W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed. KCV **E:** AEK, AEK-EK, AccessID.	IND_1

Table 4-4: Approved Services

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 C, DRBG V, Entropy Input String, Entropy Seed, KCV, SMK, 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 C, DRBG V, KCV, SMK, 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., 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
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. iKey initialized, user initialized, login required); > 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

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
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
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, 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	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. This service is not possible for the Backup configuration.	**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

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Update firmware	This service validates and then loads a new module 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 #A674) – Signature Verification, SHA (Cert #C2020) – SHA2-384 **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
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 #C2020) - SHA2-256 **Key management technique:** N/A **Authentication technique:** N/A	AEK, AEK-EK, AccessID.	Any role	**E:** AEK, AEK-EK, AccessID.	IND_1
HSM zeroize	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, 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, 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

Table 4-4: Approved Services

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 #A674) – Signature Verification, SHA (Cert #C2020) – SHA2-384 **Key management technique:** N/A **Authentication technique:** N/A	Root Certificate and Capability Signing Certificate, AEK, AEK-EK, AccessID.	HSM SO	**E:** Root Certificate, Capability 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	**E:** 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 #C2020) – HMAC-SHA2-256, SHA (Cert #C2020) – SHA2-256, SHA2-512 **Key management technique:** N/A **Authentication technique:** N/A	SALK, Secure Audit AccessID-HMAC, AEK, AEK-EK, AccessID.	Any role	**E:** SALK will be used if AU role initialised, Secure Audit AccessID-HMAC, 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 #C2020) – HMAC-SHA2-256, SHA (Cert #C2020– SHA2-256, SHA2-512 **Key management technique:** N/A **Authentication technique:** N/A	SALK, Secure Audit AccessID-HMAC, AEK, AEK-EK, AccessID.	Any role	**E:** SALK will be used if AU role initialised, Secure Audit AccessID-HMAC, AEK, AEK-EK, AccessID.	IND_1

Table 4-4: Approved Services

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 #C2020) – KWP mode with 256-bit key, KBKDF (Cert #C2020) **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.	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 #C2020) – HMAC-SHA2-256, SHA (Cert #C2020) – 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

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
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. CPV3 is supported for both import and export of SMK	**Algorithms:** CPV3: HASH_DRBG (Cert #A2125), AES (Cert #C2020) – AES-256 in KWP **Key management technique:** ESV (Cert #E97), CKG, SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125), SHA (Cert #C2020) – SHA2-512 CPV3 - KAS (Cert #A2125 – 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 C, DRBG V, Entropy Input String, Entropy Seed, Root Certificate, MIC, HOC and TUK4, TWC4, KEV_t, KCV and Cloning Transfer Key, SMK, AEK, AEK-EK, AccessID	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	**E:** DRBG C, DRBG V, Entropy Input String, Entropy Seed, Root Certificate, MIC and TUK4, TWC4, KEV_t, KCV and Cloning Transfer Key, AEK, AEK-EK, AccessID. **R/W**: SMK. **G:** KEV_t, Cloning Transfer Key, DRBG C, DRBG V, Entropy Input String, Entropy Seed. **W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed.	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 #C2020) – KWP mode with 256-bit key **Key management technique:** ESV (Cert #E97), CKG, SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125) **Authentication technique:** N/A	USK, DRBG C, DRBG V, Entropy Input String, Entropy Seed, SMK, DRBG C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID.	HSM SO, Partition CO	**E:** USK, DRBG C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID. **G/W:** SMK, DRBG C, DRBG V, Entropy Input String, Entropy Seed.	IND_1

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
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 #C2020) – SHA2-256 **Key management technique:** ESV (Cert #E97), SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125) **Authentication technique:** N/A	DRBG C, DRBG V, Entropy Input String, Entropy Seed (DRBG only used to create STM), AEK, AEK-EK, AccessID.	**Modules in zeroized State**: Any role **Initialized Module:** HSM SO	**E:** DRBG C, DRBG V, Entropy Input String, Entropy Seed (DRBG only used to create STM), AEK, AEK-EK, AccessID. **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed.	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 2-6 above> **Key management technique:** <All Key Establishment and Key Transport methods from listed in Table 2-2 and Table 2-6 above> **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

Table 4-4: Approved Services

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 #C2020) – KWP mode with 256-bit key, SHA (Cert #C2020) – SHA2-256 **Key management technique:** ESV (Cert #E97), CKG, SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125), PBKDF (Cert #A2125), KBKDF (Cert #C2020) **Authentication technique:** N/A	DRBG C, DRBG V, Entropy Input String, Entropy Seed, USK, PSK, KEK, 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 C, DRBG V, Entropy Input String, Entropy Seed, USK, PSK, KEK, PEK, AEK, AEK-EK, AccessID. **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed, 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 #C2020) – KWP mode with 256-bit key **Key management technique:** PBKDF (Cert #A2125), KBKDF (Cert #C2020), CKG **Authentication technique:** N/A	USK, PSK, KEK, 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

Table 4-4: Approved Services

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. This service is not possible for the Backup configuration.	**Algorithms:** AES (Cert #C2020) – KWP mode with 256-bit key, RSA (Cert #C2020) – RSA PKCS #1-v1.5 signature validation with SHA2-384 and 4096-bit modulus, SHA (Cert #C2020) – SHA2-256, SHA2-384 and SHA2-512 **Key management technique:** ESV (Cert #E97), SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125), KAS (Cert #A2125) – KAS1-basic with 4096-bit modulus, OneStep KDF with SHA2-512, AES (Cert #C2020) – KWP mode with 256-bit key **Authentication technique:** Single-Factor Crypto Software	DRBG C, DRBG V, Entropy Input String, Entropy Seed. HA_{PUB}, HA_{PK}, DRBG C, DRBG V, Entropy Input String, Entropy Seed. RND, K_{sess}, AEK, AEK-EK, AccessID.	HSM SO, Administrator, AU, Partition SO, Partition CO, Partition LCO, Partition CU	**E:** DRBG C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID. **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed. **W:** HA_{PUB}, HA_{PK}, **G:** RND, K_{sess}.	IND_1

Table 4-4: Approved Services

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 changed to that of the successfully authenticated role. Following login, the authentication state of the session is used check and track privileges associated with the role.	**Algorithms:** AES (Cert #C2020) – KWP mode with 256-bit key, SHA (Cert #C2020) – SHA2-256 **Key management technique:** PBKDF (Cert #A2125), KBKDF (Cert #C2020), ESV (Cert #E97), HASH_DRBG (Cert #A2125) **Authentication technique:** Memorized Secret, Multi-Factor Crypto Device	KEK, PEK (if password authentication used), PEN, 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. **G/W:** PEN. **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

Table 4-4: Approved Services

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 iKey connected to Thales Luna PED during initiation of the service.	**Algorithms:** AES (Cert #C2020) – KWP mode with 256-bit key **Key management technique:** ESV Cert #E97), CKG, SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125), ECDSA (Cert #C2020) – Key Generation and Signature Generation over curve P-521 **Authentication technique:** Single-Factor Crypto Software, Multi-Factor Crypto Device	ECC HOK, PAK, DRBG C, DRBG V, Entropy Input String, Entropy Seed, HSM-SKA-$C_{REMOTE,}$ GSK, RPV, RPV-C, RPV-K, PED-SKA-C and PED-SKA-K, AEK, AEK-EK, AccessID.	HSM SO	**E:** ECC HOK, PAK, GSK, AEK, AEK-EK, AccessID. **G:** PED-SKA-C and PED-SKA-K. **W:** RPV, RPV-C, RPV-K. **G/W**: HSM-SKA-$C_{REMOTE,}$ DRBG C, DRBG V, Entropy Input String, Entropy Seed.	IND_1

Table 4-4: Approved Services

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 #E97), CKG, SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125), KAS (Cert #A2125) – Derivation: fullUnified using curve P-521 with full key validation, key-pair generation, KDF: OneStep using SHA2-512, Key Confirmation: HMAC with 256-bit key, ECDSA (Cert #C2020) – signature generation and validation using P-521 and SHA2-512, SHA (Cert #C2020) – SHA2-512 **Authentication technique:** N/A	ECC MIC, ECC-HOC$_{PED}$, IV$_{HSM,}$ DRBG C, DRBG V, Entropy Input String, Entropy Seed, PAC, RPV-C, PED-SKA-C, PED-EKA-C HSM-SKA-K$_{REMOTE}$, HSM-EKA-C, **G:** PED Master Shared Secret, CWK$_{HSM}$, CWK$_{PED}$, DEK$_{HSM}$, DEK$_{PED}$, DMK$_{HSM,}$ AEK, AEK-EK, AccessID.	Any role	**E:** ECC MIC, ECC-HOC$_{PED}$, PAC, RPV-C, PED-SKA-C, PED-EKA-C HSM-SKA-K$_{REMOTE}$, HSM-EKA-C, AEK, AEK-EK, AccessID. **G:** HSM-EKA-K, HSM-EKA-C, PED Master Shared Secret, DEK$_{HSM}$, DEK$_{PED}$, DMK$_{HSM,}$ CWK$_{HSM}$, CWK$_{PED.}$ **W:** DEK$_{HSM}$, DEK$_{PED}$, DMK$_{HSM,}$ CWK$_{HSM}$, CWK$_{PED.}$ **G/E:** IV$_{HSM}$ **G/W**: DRBG C, DRBG V, Entropy Input String, Entropy Seed.	IND_2
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 #C2020) – KWP mode with 256-bit key, CTR mode with 256-bit key, HMAC (Cert #C2020) – HMAC-SHA2-256, SHA (Cert #C2020) – SHA2-256 **Key management technique:** N/A **Authentication technique:** N/A	DEK$_{HSM}$, DEK$_{PED}$, DMK$_{PED}$ CWK$_{HSM}$, CWK$_{PED,}$ IV$_{PED}$	Any role	**E:** DEK$_{HSM}$, DEK$_{PED}$, DMK$_{PED}$, CWK$_{HSM}$, CWK$_{PED,}$ IV$_{PED.}$	IND_2

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
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. This service is not possible for the Backup configuration.	**Algorithms:** AES (Cert #C2020) – KWP mode with 256-bit key **Key management technique:** ESV Cert #E97), CKG, SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125), RSA (Cert #C2020) – Key Generation (all supported methods and curves), ECDSA (Cert #C2020 - Key Generation (all supported methods and curves) **Authentication technique:** N/A	USK, DRBG C, DRBG V, Entropy Input String, Entropy Seed, Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), CITS-DAK, CITS-DAC, ECC DAC, ECC DAK, DRBG C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID.	HSM SO, Administrator (for admin partition keys) Partition CO, Partition LCO (for user partition)	**E:** USK, DRBG C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID. **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed. **W:** Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys). **E/W:** CITS-DAK, CITS-DAC, ECC DAC, ECC DAK.	IND_1
Generate domain parameters[17]	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 #C2020) – KWP mode with 256-bit key **Key management technique:** ESV Cert #E97), CKG, SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125), DSA (Cert #C2020) – Domain parameter generation **Authentication technique:** N/A	DRBG C, DRBG V, Entropy Input String, Entropy Seed. , AEK, AEK-EK, AccessID.	Any role	**E:** DRBG C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID. **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed.	IND_1

[17] Public users cannot generate any objects where either CKA_SENSITIVE or CKA_PRIVATE attributes are true.

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
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. This service is not possible for the Backup configuration.	**Algorithms:** AES (Cert #C2020) – KWP mode with 256-bit key **Key management technique:** ESV Cert #E97), SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125), KBKDF (Cert #C2020), KAS-ECC-SSC (Cert #A2125), KAS-FFC-SSC (Cert #A2125), KDA (Cert #A2125), CVL (Cert #A2125) – X9.42 KDF **Authentication technique:** N/A	Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys), USK, DRBG C, DRBG V, Entropy Input String, Entropy Seed. Symmetric Keys (general partition or session keys), DRBG C, DRBG V, Entropy Input String, Entropy Seed, 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 C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID. **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed. **W:** Symmetric Keys (general partition or session keys).	IND_1
Import public key, certificate, domain object or data objects[18]	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. This service is not possible for the Backup configuration.	**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

[18] Public users cannot generate/import any objects where either CKA_SENSITIVE or CKA_PRIVATE attributes are true.

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
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.	**Algorithms:** AES (Cert #C2020) – KWP mode with 256-bit key **Key management technique:** KTS-IFC (Cert #A2125) – KTS-RSA-OAEP-basic with any supported key size, SHA (Cert #C2020) – any supported hash, RSA (CVL Cert #C2020) – RSA PKCS #1 v1.5 wrapping, AES (Cert #C2020) – all supported key sizes and modes ECB, CBC, CTR, KW, KWP, Triple-DES (Cert #C2020) – 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

Table 4-4: Approved Services

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. This service is not possible for the Backup configuration.	**Algorithms:** AES (Cert #C2020) – KWP mode with 256-bit key **Key management technique:** ESV Cert #E97), SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125), KTS-IFC (Cert #A2125) – KTS-RSA-OAEP-basic with any supported key size, RSA (CVL Cert #C2020) – RSA PKCS #1 v1.5 wrapping, AES (Cert #C2020) – 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 C, DRBG V, Entropy Input String, Entropy Seed, USK. , 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 C, DRBG V, Entropy Input String, Entropy Seed, USK, AEK, AEK-EK, AccessID. **R**: Asymmetric Key Pairs (general partition or session keys), Symmetric Keys (general partition or session keys). **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed.	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

Table 4-4: Approved Services

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 #C2020) – KWP mode with 256-bit key **Key management technique:** ESV Cert #E97), HASH_DRBG (Cert #A2125) SIM2+ Format Objects: AES (Cert #C2020) – GCM mode with 256-bit key SIM3 Format Objects: AES (Cert #C2020) – CBC mode with 256-bit key, SHA (Cert #C2020) – SHA2-256 SIM2 Format Objects: AES (Cert #C2020) – CTR mode with 256-bit key, SHA (Cert #C2020) – 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, ESV, DRBG C, DRBG V, Entropy Input String, Entropy Seed.	HSM SO, Administrator, Partition CO, Partition LCO, Partition CU	**E:** SMK, USK, AEK, AEK-EK, AccessID. **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed. **W:** Symmetric Keys (general partition or session keys) or Asymmetric Key Pairs (general partition or session keys).	IND_1

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
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 #C2020) – KWP mode with 256-bit key, HMAC (Cert #C2020)– HMAC-SHA2-256, SHA (Cert #C2020) –SHA2-256, SHA2-384 **Key management technique:** AES (Cert #C2020) – 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
Cryptographic Services						
Re-seed partition DRBG	This service is used by a user to trigger a manual re-seed operation of the DRBG.	**Algorithms:** N/A **Key management technique:** ESV Cert #E97), SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125) **Authentication technique:** N/A	DRBG C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU	**E/G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed. **E:** AEK, AEK-EK, AccessID.	IND_1
Extract entropy from DRBG	This service is used by a user to request and export entropy from the DRBG.	**Algorithms:** N/A **Key management technique:** ESV Cert #E97), HASH_DRBG (Cert #A2125 **Authentication technique:** N/A	DRBG C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID.	HSM SO, Administrator, Partition SO, Partition CO, Partition LCO, Partition CU	**E/G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed. **E:** AEK, AEK-EK, AccessID.	IND_1

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
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.	**Algorithms:** SHA (Cert #C2020) – all hash options supported, SHA3 (Cert #C2020) – 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
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.	**Algorithms:** AES (Cert #C2020) – all supported modes and key sizes. **Key management technique:** ESV Cert #E97), SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125) – 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 C, DRBG V, Entropy Input String, Entropy Seed, 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 C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID. **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed.	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.	**Algorithms:** AES (Cert #C2020) – all supported modes and key sizes, Triple-DES (Cert #C2020) – 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

Table 4-4: Approved Services

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. This service is not possible for the Backup configuration.	**Algorithms:** RSA (Cert #C2020), ECDSA (Cert #C2020), DSA (Cert #C2020), HMAC (Cert #C2020), CMAC (Cert #C2020) – AES and Triple-DES, AES (Cert #C2020) – KWP mode with 256-bit key **Key management technique:** ESV Cert #E97), SHA (Cert #C2020) – SHA2-512, HASH_DRBG (Cert #A2125) **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 C, DRBG V, Entropy Input String, Entropy Seed, 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 C, DRBG V, Entropy Input String, Entropy Seed, AEK, AEK-EK, AccessID. **G/W:** DRBG C, DRBG V, Entropy Input String, Entropy Seed	IND_1
Verify 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.	**Algorithms:** RSA (Cert #C2020), ECDSA (Cert #C2020), DSA (Cert #C2020), HMAC (Cert #C2020), CMAC (Cert #C2020) – AES and Triple-DES, AES (Cert #C2020) – 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, 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

Table 4-4: Approved Services

Service	Description	Approved Security Functions	Key and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Bootloader Services						
Request complete erase of the HSM main firmware image and key stores (excludes erase of bootloader)	This service is used to recover from corrupt main firmware and is performed as a factory operation. Following erase, 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
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 #C2022) – RSA PKCS #1 v1.5 signature validation with modulus length 4096, SHA (Cert #C2022) – 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

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 13.3 and 13.4).

As notes on the content of Table 4-5:

> In the ‘Indicator Column’:

- IND_1:
  - **For USB HSM** – **Partition Policy (43) Allow non-FIPS Algorithms** is set to `disabled` AND **HSM Policy (56), Allow User Defined ECC Curves** is set to `disabled` AND return code is `CKR_OK`; or
  - **For Backup HSM** – **HSM Policy (55), Enable Restricted Restore** is set to `enabled` AND return code is `CKR_OK`.

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[19], 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[20], RSA X.509, SEED, SM4, Triple-DES[21], 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[22], 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

[19] RSA is non-compliant when using PKCS#1, v1.5 padding for encryption or decryption.
[20] RSA is non-compliant with less than 112 bits of encryption strength.
[21] Triple-DES is non-compliant with less than 112 bits of encryption strength.
[22] 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[23], ECDSA[24], EdDSA, EdDSA PH, KCDSA, RSA[25], SM2, SM3.		
Verify 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[26], HMAC[27], 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[28], ECDSA[29], EdDSA, EdDSA PH, KCDSA, RSA[30], SM2, SM3.	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 to other cryptographic modules or systems. This service is not possible for the Backup configuration.	AES[31], ARIA, BIP32, DES, MD5, SHA, SSL PRE-MASTER, SSL3-MASTER, SM3, Triple-DES, XOR.	HSM SO, Administrator, Partition CO, Partition LCO.	IND_1

[23] DSA is non-compliant with less than 112 bits of encryption strength.
[24] ECDSA is non-compliant with less than 112 bits of encryption strength.
[25] RSA is non-compliant with less than 112 bits of encryption strength.
[26] Triple-DES-CMAC is non-compliant with less than 112-bits of encryption strength.
[27] HMAC is non-compliant with less than 112-bits of encryption strength.
[28] DSA is non-compliant with less than 112 bits of encryption strength.
[29] ECDSA is non-compliant with less than 112 bits of encryption strength.
[30] RSA is non-compliant with less than 112 bits of encryption strength.
[31] 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
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[32], ECC[33], KCDSA, RSA[34], 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[35], 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[36], ARIA, CAST3, CAST5, DES, RC2, RSA, SEED, SM4, TDES.	HSM SO, Administrator, Partition CO, Partition LCO.	IND_1
Generate domain parameters.[37]	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	X9.42 Domain Parameter Generation	Any role.	IND_1

[32] Diffie-Hellman key generation mechanisms are non-compliant with less than 112-bits of encryption strength.
[33] ECC key generation mechanisms are non-compliant with less than 112-bits of encryption strength.
[34] RSA key generation mechanisms are non-compliant with less than 112-bits of encryption strength.
[35] 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.
[36] 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.
[37] 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.

Table 4-5: Non-Approved Services

Service	Description	Non-Approved Algorithms Accessed	Roles	Indicator
	cryptographic services or to export to other cryptographic modules or systems.			

5 Software/Firmware Security

5.1 Firmware Integrity

The Thales Luna G7 Cryptographic Module's firmware integrity is checked on startup as described in section 10.1. The two keys used for the software and firmware Approved integrity techniques are the Root Certificate and Firmware Signing Certificate.

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 bootloader and main 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 13.5.

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 Firmware Load

When new 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 firmware is a two stage process. The first stage is to download the firmware to the module. The second stage involves subsequently re-authenticating and loading the 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.

5.3 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.

6 Operational Environment

The module uses a limited modifiable operational environment.

7 Physical Security

7.1 Mechanism Summary

7.1.1 Module Construction

The module is enclosed in a plastic 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 plastic 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 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 Following any extended periods of unattended storage for the module	<see below>.

Following manufacture, both the front and rear covers of the Thales Luna G7 Cryptographic Module are permanently adhered to the PCB assembly using epoxy resin and with the lid assemblies having feet set into the epoxy.

The polycarbonate case will show witness marks if any attempt is made to tamper with the plastic.

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

In the event of any observed damage, 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 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 the 5V voltage rails that 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	0°C ± 2°C	EFP	shutdown
High Temperature	+70°C ± 2°C	EFP	shutdown
Low Voltage	3.9V ± 0.11V	EFP	shutdown
High Voltage	5.71V ± 0.145V	EFP	shutdown

7.4 Module Case and Coatings

The module PCB is potted using an epoxy-based compound inside the polycarbonate plastic case that makes up the identified cryptographic boundary in Figure 2-1 and Figure 2-2.

The potting material has a Shore D hardness rating of 90 and an operating temperature from -65ºC to +200ºC over which it maintains its hardness.

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

8 Non-invasive security

N/A: [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 (Explicit zeroization);
- KDM2: RAM reset (Implicit zeroization);
- KDM3: Erasure of entire memory sector(s) (Explicit zeroization); and
- KDM4: Erasure of the encrypting keys (Implicit zeroization).

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.

NOTE When an end-user triggers zeroization using a zeroization service and zeroization is successful, return code `CKR_OK` is displayed.

> 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
Root Certificate 4096-bit public key certificate	150-bits	RSA SigVer (#A674)	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-bits	RSA SigVer (#A674)	N/A – generated outside the module.	Input with Firmware Update File, which is considered plaintext	N/A	Stored on Thales managed "Engineering" HSM	Management Policies	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-bits	RSA SigVer (#A674)	N/A – generated outside the module.	Input with Capability Update File	N/A	Stored on Thales managed "Engineering" HSM	Management Policies	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-bits	RSA SigVer (#A674)	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-bits	ECDSA SigVer (Cert #C2020)	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-bits	RSA SigGen (#A674)	[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 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-bits	RSA SigVer (#A674)	[FIPS 186-4], Appendix B.3.6	Certificate Output in Plaintext	N/A	Flash memory in plaintext	Full HSM Wipe - 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.**
ECC Hardware Origin Key (ECC HOK) ECC private key on curve P-384	192-bits	ECDSA SigGen (Cert #C2020)	[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.**
ECC Hardware Origin Certificate (ECC HOCtw) ECC public certificate for public key on curve P-384	192-bits	ECDSA SigVer (Cert #C2020)	[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 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 (TUK4) 4096-bit private key	150-bits	RSA SigGen (#A674)	[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-bit public key certificate	150-bits	RSA SigVer (#A674)	[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 (KEV_t) as part of the handshake during the cloning protocol. **This key is a PSP.**
Cloning Key Encryption Vector – target (KEV_t) 384 bit nonce	256-bits	KDA One-Step Sp800-56Cr2 (Cert #A2125)	[SP800-90Ar1] HASH_DRBG with SHA2-256	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 target 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
Cloning Transfer Key 256-bit AES key	256-bits	AES-CBC (Cert #C2020)	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.**
Device Authentication Key (CITS-DAK) 4096-bit private key	150-bits	RSA SigGen (#A674)	[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.**
Device Authentication Key (CITS-DAC) 4096-bit public key certificate	150-bits	RSA SigVer (#A674)	[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-bits	ECDSA SigGen (Cert #C2020)	[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-bits	ECDSA SigVer (Cert #C2020)	N/A – generated outside the module.	Loaded in plaintext during manufacturing 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.**
Entropy Input String	384-bits	HASH_DRBG (Cert #A2125)	Full-entropy conditioned output from ESV Cert #E97) 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 [SP800-90Ar1] Hash_DRBG using SHA2-256 as the PRF. **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
Entropy Seed	640-bits	HASH_DRBG (Cert #A2125)	N/A	Not Input or Output	N/A	Working SDRAM in plaintext	Power Cycle - KDM2	640-bit value that is the concatenation of the Entropy Input String (384-bits), Nonce (128-bits) and 128-bit personalisation string. Used as to initialise the internal state of HASH_DRBG. **This key is a CSP.**
DRBG V	256-bits	HASH_DRBG (Cert #A2125)	Internal state generated using HASH_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.**
DRBG C	256-bits	HASH_DRBG (Cert #A2125)	Internal Constant value	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-bits	AES-KWP (Cert #C2020)	[SP800-90Ar1] HASH_DRBG with SHA2-256	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.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
Role Domain Key (RDK) 256-bit key	256-bits Or 32 to 2040-bits	KDA One-Step Sp800-56Cr2 (Cert #A2125)	[SP800-90Ar1] HASH_DRBG with **SHA2**-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-bits	HMAC-SHA2-256 (Cert #C2020)	[SP800-90Ar1] HASH_DRBG with SHA2-256	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-bits	HMAC-SHA2-256 (Cert #C2020)	[SP800-90Ar1] HASH_DRBG with SHA2-256	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.**
User Password (if PED configuration and optionally selected) 8 - 255 character data string	32 to 256-bits	PBKDF (Cert #C2020)	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.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
AccessID Encryption Key – Encryption Key (AEK-EK) 256-bit AES key	256-bit.	AES-KW (Cert #C2020).	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-KWP (Cert #C2020).	[SP800-90Ar1] HASH_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-bits	N/A	SHA2-256 (Cert #C2020)	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-bits	KBKDF (Cert #C2020)	[SP800-90Ar1] HASH_DRBG with SHA2-256	Input / Output via direct connection to PED All messages sent to the PED are encrypted using HSM CSP Wrapping Key and AES KWP	N/A	Not stored on module	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-bits	PBKDF (Cert #A2125)	N/A	Input from host using ICD communication path and encrypted under the PEC and using KTS-OAEP-basic from [SP800-56Br2]	N/A	Not stored on module	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-bits	AES-KWP (Cert #C2020)	[SP800-90Ar1] HASH_DRBG with SHA2-256	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-bits	AES-KWP (Cert #C2020)	[SP800-90Ar1] HASH_DRBG with SHA2-256	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-bits	AES-KWP (Cert #C2020)	[SP800-90Ar1] HASH_DRBG with SHA2-256	Input/Output using CPV3	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-bits	KAS-IFC (Cert #A2125)	[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-bits	KAS-IFC (Cert #A2125)	[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-bits	AES-KWP (Cert #C2020)	[SP800-90Ar1] HASH_DRBG with SHA2-256	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-bit AES key	256-bits	AES-KWP (Cert #C2020)	[SP800-90Ar1] HASH_DRBG with SHA2-256	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-bits	KDA One-Step Sp800-56Cr2 (Cert #A2125)	[SP800-90Ar1] HASH_DRBG with SHA2-256 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 an iKey to enable initializing additional modules into the domain. **This key is a CSP.**
Remote PED Vector (RPV) (if PED configuration)	256-bits	KDA One-Step Sp800-56Cr2 (Cert #A2125)	[SP800-90Ar1] HASH_DRBG with SHA2-256	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.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
PED Authentication Certificate (PAC) ECC public key on curve P-521	256-bits	ECDSA SigVer (Cert #C2020)	[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 remote connection with a Luna PED. **This key is a PSP.**
PED Authentication Key (PAK) ECC private key on curve P-521	256-bits	ECDSA SigGen (Cert #C2020)	[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 Remote Connections (HSM-SKA-C_{REMOTE}) ECC public key on curve P-521	256-bits	KAS-ECC (Cert #A2125)	[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, KAS-ECC) key-agreement for remote connection with PED. **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-bits	KAS-ECC (Cert #A2125)	[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, KAS-ECC) key-agreement agreement for remote connection with PED. **This key is a CSP.**
HSM Ephemeral Key-Agreement Certificate (HSM-EKA-C) ECC public key on curve P-521	256-bits	KAS-ECC (Cert #A2125)	[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, KAS-ECC) key-agreement agreement for remote connection with a Thales 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
HSM Ephemeral Key-Agreement Private Key (HSM-EKA-K) ECC private key on curve P-521	256-bits	KAS-ECC (Cert #A2125)	[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, KAS-ECC) 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-bits	ECDSA SigVer (Cert #C2020)	[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-bits	ECDSA SigGen (Cert #C2020)	[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-bits	KAS-ECC (Cert #A2125)	[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, KAS-ECC) 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-bits	KAS-ECC (Cert #A2125)	[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, KAS-ECC) 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-bits	KDA One-Step Sp800-56Cr2 (Cert #A2125)	N/A	N.A	KAS-ECC (Cert #A2125)	Working SRAM in plaintext.	Zeroized following use – KDM1	Intermediate key value used during setup of the 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.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
HSM CSP Wrapping Key (CWK_{HSM}) 256-bit AES key	256-bits	AES-KWP (Cert #C2020) AES-CTR (Cert #C2020)	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 wrapping exchanged SSPs. Key is used with either AES-KWP or AES-CTR, depending on the cipher suite negotiated between the HSM and PED during remote PED setup. **This key is a CSP.**
PED CSP Wrapping Key (CWK_{PED}) 256-bit AES key	256-bits	AES-KWP (Cert #C2020) AES-CTR (Cert #C2020)	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 wrapping exchanged SSPs. Key is used with either AES-KWP or AES-CTR, depending on the cipher suite negotiated between the HSM and PED during remote PED setup. **This key is a CSP.**
HSM Data Encryption Key (DEK_{HSM}) 256-bit AES key	256-bits	AES-KWP (Cert #C2020) AES-CTR (Cert #C2020)	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). Key is used with either AES-KWP or AES-CTR, depending on the cipher suite negotiated between the HSM and PED during remote PED setup. **This key is a CSP.**
HSM MAC Key (DMK_{HSM}) 256-bit HMAC key	256-bits	HMAC-SHA2-256 (#C2020)	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.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
HSM Initialization Vector (IV_{HSM}) 256-bit IV	256-bits	AES-KWP (Cert #C2020) AES-CTR (Cert #C2020)	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). Key is used with either AES-KWP or AES-CTR, depending on the cipher suite negotiated between the HSM and PED during remote PED setup. **This key is a CSP.**
PED Data Encryption Key (DEK_{PED}) 256-bit AES key	256-bits	AES-KWP (Cert #C2020) AES-CTR (Cert #C2020)	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). Key is used with either AES-KWP or AES-CTR, depending on the cipher suite negotiated between the HSM and PED during remote PED setup. **This key is a CSP.**
PED MAC Key (DMK_{PED}) 256-bit HMAC key	256-bits	HMAC-SHA2-256 (#C2020)	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-bits	AES-KWP (Cert #C2020)	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.**

Key / SSP Name / Type	Strength	Security Function and Cert Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use and Related Keys
Password Encryption Key (PEK) 4096 bit private key	150-bits	RSA SigGen (Cert #A674)	[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-bits	RSA SigVer(Cert #A674)	[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.**
Password Encryption Nonce (PEN) 192-bit nonce	192-bits	N/A	[SP800-90Ar1] HASH_DRBG with SHA2-256	Output in Plaintext Imported encrypted under the PEC using KTS-OAEP.	N/A	Working RAM in plaintext	Zeroized following use – KDM1.	Nonce used to provide replay protection with the PEC protocol. **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
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 SigGen RSA SigVer (Cert #C2020, #A674) ECDSA SigGen ECDSA SigVer (Cert #C2020, #A2125) DSA SigGen DSA SigVer (Cert #C2020) KAS (KAS-ECC-SSC (Cert #A2125) and KDA (Cert #A2125)) KAS (KAS-ECC-SSC (Cert #A2125) and CVL (Cert #A2125)) KAS (KAS-FFC-SSC (Cert #A2125) and KDA (Cert #A2125)) KAS (KAS-FFC-SSC (Cert #A2125) and CVL (Cert #A2125)) KAS-IFC (Cert #A2125) KTS-IFC (Cert #A2125)	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.** **This key is a PSP.**

Symmetric Keys (general partition or session keys) AES or Triple-DES, MAC, KDF	128, 192 or 256-bit for AES keys. 112-bit for Triple-DES.	Triple-DES-CBC Triple-DES-CFB64 Triple-DES-CFB8 Triple-DES-CMAC Triple-DES-CTR Triple-DES-ECB Triple-DES-OFB (Cert #C2020) 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 (Cert #C2020) KDA (Cert #A2125) KBKDF (Cert #C2020) AES-CBC AES-CFB128 AES- CFB8 AES-CMAC AES-CTR AES-ECB AES-GCM AES-KW AES-KWP AES-OFB (Cert #C2020)	N/A (user imported) Or [SP800-90Ar1] HASH_DRBG with SHA2-256 (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	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 asymmetric key pairs that can be exported/imported from/to the module or generated by the 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
				partitions using SKS, encrypted under the SMK				

9.2 Non-Deterministic Random Number Generation Specification

The module includes a non-deterministic Random Number Generator (RNG) within the module boundary.

The non-deterministic RNG is used exclusively to feed the DRBG (Cert #C2020).

The Non-Deterministic RNG complies with [SP800-90B] and has been certified using [FIPS 140-3 IG] D.J with guidance set out in [FIPS 140-3 IG] D.K.

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

Entropy sources	Minimum number of bits of entropy	Details
Non-deterministic jitter from FRO.	The noise source outputs blocks of entropy in 384 bits with H = 0.838411. Following testing, the DRBG is seeded with a 256-bit seed and 128-bit nonce from the noise source containing 384 * 0.838411 which equals 321 bits of entropy.	[SP800-90B] compliant Non-Deterministic RNG using a hardware based noise internal to the module boundary (ESV #E97). Raw noise collection is performed autonomously by hardware as entropy is required to seed the on-chip DRBG (Cert #C2020). If the entropy register is not full when the DRBG accesses it, the read will stall until the entropy is generated. During each entropy collection cycle, 2500 samples of raw noise are collected. All outputs from the noise source are subjected to statistical testing ahead of being fed to the DRBG. 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:

> **Manual Key Import/Export**

The manual import/export methods of the module are restricted to the G7 GUI and the iKey directly attached to the module via the USB port.

- For the G7 GUI the import methods include password authentication data. This includes whether the module is using Password authentication or is optionally selected with PED authentication.
- For the iKey the import/export methods include the PED authentication data, private/public keys and PED Key Agreement certificates.

> **Password Encryption Certificate (PEC) Import Protocol**

Prior to submitting a password for authentication, the client:

- requests a 24-byte nonce using the LUNA_REQUEST_CHALLENGE command;
- requests the cryptographic module PEC using the LUNA_GET_PEC command; and
- validates the received certificate against an embedded copy of the Thales root certificate stored in the client.

Following completion of above – the client concatenates the nonce (24 bytes) with their password (up to 248 bytes) and encrypts using RSA-OAEP with SHA512 for its MGF as defined in [SP800-56Br2] as KTS-OAEP-basic.

> **Plaintext Certificate Import**

There are various plaintext certificates that can be imported onto the module via the ICD interface. These include:

- Capability Signing Certificate, which is imported along with the Capability Update File (CUF).
- Firmware Signing Certificate, which is imported along with the Firmware Update File (FUF).
- HA Login Public Key which is a 4096-bit RSA public key used for the HA Login protocol.

For more information on the certificates, refer to Table 9-1 above.

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.

This service is not possible for the Backup configuration.

> **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 TWC4 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 CPV3.

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: SHA1, 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.

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.

10 Self-Tests

10.1 Pre-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
Boot loader performs an RSA PKCS #1-v1.5 signature with 4096-bit modulus and SHA2-384 signature verification of itself.	Verify, 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 prior to firmware start.	Verify, 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 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

Test	Cryptographic Mechanism Tested	Location	When Performed	Operations Performed	Indicator
Cryptographic Algorithm Self-Test (CAST)					
SHA KAT (#C2022)	**Pre-operational:** SHA2-384 **PST:** N/A	Bootloader	Prior to first use.	Digest.	Error output and module halt
RSA KAT (#A6549)	**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
Diffie-Hellman - FFC full public-key validation ([SP800-56Ar3]/[SP800-56Br2] compliant implementation)	Tested as part of all key agreement operations.	Main firmware	Ahead of public key use for derive operation.	Derive.	Error output
ECDH – ECC full public-key validation ([SP800-56Ar3] compliant implementation)	Tested as part of all key agreement operations.	Main firmware	Ahead of public key use for derive operation.	Derive.	Error output
DRBG KAT (#C2020)	**Pre-operational:** Instantiate, Generate and Re-seed KAT for HASH_DRBG with SHA-256 **PST:** *<as per pre-operational self-test>*	Main firmware	Prior to first use., PST	N/A.	Error output and module halt
SHA KAT (#C2020)	**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	Main firmware	Prior to first use., PST	Digest.	Error output and module halt

Test	Cryptographic Mechanism Tested	Location	When Performed	Operations Performed	Indicator
HMAC KAT (#C2020)	**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	Main firmware	Prior to first use., PST	Digest.	Error output and module halt
RSA KAT (#C2020 and #A674).	**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 with SHA2-256 and SHA2-256 for MGF, 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	Main firmware	Prior to first use., PST	Sign, Verify, Encrypt, and Decrypt.	Error output and module halt
DSA KAT (Signature Generation, Sig Verification) (#C2020)	**Pre-operational:** Signature Generation, Signature Verification for 2048-bit modulus with SHA2-224. Signature Verification with 1024-bit modulus and SHA1 **PST:** DSA with 2048-bit modulus and SHA2-224	Main firmware	Prior to first use., PST	Sign and Verify.	Error output and module halt
Diffie-Hellman KAT (Key Agreement only, No Derivation) (#A2125)	**Pre-operational:** X9.42 Diffie-Hellman [SP800-56Ar3] key derive with 2048-bit modulus **PST:** *<as per pre-operational self-test>*	Main firmware	Prior to first use., PST	Derive (X9.42 Derive operation – no KDF).	Error output and module halt
AES KAT (#C2020)	**Pre-operational:** ECB, CBC, OFB, CFB128, CFB8, KW, KWP, GCM and CMAC covering 128-bit,192-bit and 256-bit keys as supported by the different modes **PST:** CBC, GCM, KW, KWP – 256-bit key	Main firmware	Prior to first use., PST	Encrypt, Decrypt.	Error output and module halt

Test	Cryptographic Mechanism Tested	Location	When Performed	Operations Performed	Indicator
Triple-DES KAT (#C2020)	**Pre-operational:** ECB, CBC, OFB, CFB64, CTR 168-bit keys. **PST:** ECB with 168-bit key	Main firmware	Prior to first use., PST	Decrypt.	Error output and module halt
Triple-DES KAT (#C2020)	**Pre-operational:** CMAC for 168-bit keys. **PST:** *<as per pre-operational self-test>*	Main firmware	Prior to first use., PST	Verify.	Error output and module halt
ECDH KAT (#A2125)	**Pre-operational:** KAS-ECC [SP800-56Ar3] shared secret calculation (only) using curves P-224, P-384, P-521 and K-233 **PST:** KAS-ECC with P-384, OneStep KDF using SHA2-256	Main firmware	Pre-operational, PST	Derive (no KDF).	Error output and module halt
ECDSA KAT (#C2020)	**Pre-operational:** Signature Generation, Signature Verification with ECDSA and both curves P-256 and K-233 (no hashing) **PST:** Signature Generation, Signature Verification using ECDSA and curves P-256, K-233	Main firmware	Prior to first use., PST	Sign, Verify.	Error output and module halt
KBKDF KAT (#C2020)	**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	Main firmware	Prior to first use., PST	Derive.	Error output and module halt
KDF KAT (#C2020)	**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 and X9.63 KDF [SP800-135r1] using SHA1**.**	Main firmware	Prior to first use., PST	Derive.	Error output and module halt
KAS1-basic KAT (#A2125)	**Pre-operational:** KAS1-basic [SP800-56Br2] with 4096-bit modulus **PST:** *<as per pre-operational self-test>.*	Main firmware	Prior to first use., PST	Encrypt, Decrypt.	Error output and module halt

Test	Cryptographic Mechanism Tested	Location	When Performed	Operations Performed	Indicator
PBKDF KAT (#A2125)	**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	Main firmware	Prior to first use., PST	Derive.	Error output and module halt
Pair-Wise Consistency Test (PCT)					
RSA PCT	Performed for all RSA key generation mechanism	Main firmware	On generation	Encrypt, Decrypt, Sign, Verify.	Error output and module halt
DSA PCT	Performed for all DSA key generation mechanism	Main 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	Main firmware	On generation	Sign, Verify, and Derive.	Error output and module halt
Software/Firmware Load Test (SW/FW Load)					
Firmware Load Test	**Continuous Test:** RSA PKCS #1-v1.5, modulus 4096 signature and SHA2-384	Main firmware	On firmware update request	Verify.	Error output and FW update request rejected
Manual Entry Test					
N/A					
Bypass Test					
N/A					
Critical Function Test					
HRNG conditional tests	**Continuous Test:** Total failure test on the output from the hardware noise source, Repetition Count Test and Adaptive Proportion Test statistical tests	Main firmware	Continuous	N/A.	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 G7 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 defence-in-depth approach to security.

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

- securing its location using physical defences; 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

Before 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 the module should be received in the zeroised state with no initialized roles.

NOTE Failing to follow the steps to configure the module into the Approved Mode will result in the module operating in a Non-Compliant state, which is outside the scope of this validation.

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 13.1.

The module is confirmed as being in the zeroised state when the `partition status` for the administration slot reports `zeroized`.

Initialization creates the HSM SO role, names the module, defines the authentication mode and associates the admin partition with a key cloning domain.

Initialization is performed using the `hsm init` command from LunaCM or LunaSH, though LunaSH can only be used for the Thales Luna G7 Backup 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 13.3 and 13.4.

NOTE As part of initialization when using PED based authentication, the end-user is asked if they wish to duplicate your iKeys. 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 iKeys 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 iKeys 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, though encrypted, should be maintained on a 'need to know' basis in order to minimize un-necessary creation of copies of the encrypted key object. 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: iKeys.

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 G7 Cryptographic Module to be read by the host driver even if the bootloader or main 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 LunaCM 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 main 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 SHA2-224 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 iKeys (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.

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 iKeys

11.6.1 User Authentication iKeys

Should an end user lose an 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 iKey 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 iKey.
  - 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 iKey is lost, a copy of the Audit Logging Secret should be made and transferred to a PED iKey.
  - For the loss of an HSM SO iKey, LunaCM command `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 iKey, 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 LunaCM command `role init –name audit` to recreate a new AU iKey. The Audit Logging Secret can then be imported using the `audit import` command 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 iKey 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 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:

- run the **`ped vector init`** from LunaCM.

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 an iKey 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:

> 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 main 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 main firmware in a format not made publicly available can be loaded onto the module.

11.10 Resetting the HSM

Resetting is the process of removing all sensitive information from the cryptographic module.

Run the LunaCM **`hsm factoryreset`** command to reset 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

11.11 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 fwupdateG7_testCert_7.0.1_RC327.fuf -
authcode fwupdateG7_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 11-2: Example successful FW update console output from LunaCM

NOTE All updates of the firmware MUST be FIPS 140-3 validated before they can be loaded for the module to remain in an approved mode of operation.

11.12 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 and 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. Both hardware part numbers in this security policy are correlated with the Thales Luna G7 Cryptographic Module.

Checking the module's name, hardware, bootloader and main firmware versions with[38]:

> `hsm showinfo` when using LunaCM.

The command returns status information on the target cryptographic module including the version numbers for both bootloader, main firmware and separately the hardware identity. Example output for the command for a valid module is shown in the figure below with relevant versions and the module's name highlighted in red:

```
lunacm:>hsm showinfo

        Slot Id -> 6
        Partition Label -> g7_597068
        Partition Serial Number -> 597068
        Partition Model -> Luna G7
        Partition Manufacturer -> Gemalto
        Partition Status -> L3 Device, OK[39]
        Session State -> CKS_RW_PUBLIC_SESSION
        Role Status ->   none logged in
        RPV Initialized -> No

        Partition SMK OUIDs:
                SMK-FW4: Not Initialized
                SMK-FW6: Not Initialized
                SMK-FW7-FM: Not Initialized
                SMK-FW7-Rollover: Not Initialized
                SMK-FW7-Primary: 3c000000170000124c1c0900
```

[38] LunaCM maps to the Luna ICD logical interface at the cryptographic module boundary.
[39] The command will show "Zeroized" instead of "OK" if the module is in a zeroized state.

```
        Partition Storage:
                Total Storage Space:  655360
                Used Storage Space:   0
                Free Storage Space:   655360
                Object Count:         0
                Overhead:             24408

        HSM Storage:
                Total Storage Space:  33554432
                Used Storage Space:   679768
                Free Storage Space:   32874664
                Allowed Partitions:   1
                Number of Partitions: 0

        HSM Part Number -> 808-000080-001

        Environmental:
                System Temperature :  35 deg. C

        Firmware Version -> 7.7.3
        Bootloader Version ->  1.6.0
        Rollback Firmware Version -> Not Available

        License Count:
                1. 621000186-000 G7 Base CUF

        *** The HSM is in FIPS approved operation mode. ***

Command Result : No Error
```

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

13.2 Identifying the Module Type

The Thales Luna G7 Cryptographic Module comes in two configurations, the Thales Luna G7 USB HSM and the Thales Luna G7 Backup HSM. Check the configuration with:

> **`slot list`** when using LunaCM.

The command returns status information of the cryptographic module including the configurations. Example output for the command for a valid module is shown in the figure below with relevant configurations highlighted in red:

```
Available HSMs:
        Slot Id ->              105
        Label ->                g7backup_596424
```

```
        Serial Number ->        596424
        Model ->                Luna G7
        Firmware Version ->     7.7.3
        Bootloader Version ->   1.6.0
        Configuration ->        Luna HSM Admin Partition (PW) Backup Mode
        Slot Description ->     Admin Token Slot
        HSM Status ->           L3 Device, OK

        Slot Id ->              108
        Label ->                g7_121212
        Serial Number ->        121212
        Model ->                Luna G7
        Firmware Version ->     7.7.3
        Bootloader Version ->   1.6.0
        Configuration ->        Luna HSM Admin Partition (PW) Key Export With
Cloning Mode
        Slot Description ->     Admin Token Slot
        HSM Status ->           L3 Device, OK
```

13-2: Example output of `slot list` command from LunaCM

13.3 Approved Mode of Operation for USB HSM

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 (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** is **`enabled`** by default and shall be **`disabled`**.

If the HSM SO attempts to enable or disable these policies, a warning is displayed and the HSM SO is prompted to confirm the selection. If this policy is left as **`enabled`**, the module will be operating in the non-approved mode of operation.

Following entry into an approved mode of operation, any changes to either policy will trigger an automatic zeroization of the HSM erasing all roles and partition stored key objects.

13.4 Approved Mode of Operation for Backup HSM

To place the Thales Luna G7 Backup HSM into its approved mode of operation, the HSM SO must check and, if necessary, set the following HSM level policy:

> **HSM Policy (55), Enable Restricted Restore** – this is **`disabled`** by default and shall be **`enabled`**.

If the HSM SO attempts to disable these policies, a warning is displayed and the HSM SO is prompted to confirm the selection. If this policy is left as **disabled**, the module will be operating in the non-approved mode of operation.

Following entry into an approved mode of operation, any changes to **HSM Policy (55), Enable Restricted Restore** will trigger an automatic zeroization of the HSM erasing all roles and partition stored key objects.

13.5 Using CA_PerformSelftest

To make the module perform its cryptographic self-tests as described in section 10.2 you must use the following:

> **Self Test** (Option #90) when using the 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

13.6 Nominal Ranges

The nominal operating temperatures of the module are between 5 and 35°C. The module's tamper thresholds can be found in section 7.3.

The nominal operation voltage of the module is 5V DC. The module's voltage thresholds can also be found in section 7.3.

13.7 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, as shown above in section 13.2. 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

13.8 Additional Guidance

In addition to the direct guidance provided in this Security Policy, both Thales Luna G7 USB HSM and Thales Luna G7 Backup 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 Thales Luna G7 USB HSM. Thales Luna G7 Backup HSM documentation is part of the Luna PCIe HSM and Luna Network HSM documentations.

As part of the product documentation:

- **HSM Administration Guide** – describes how to install your Thales Luna G7 USB HSM 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.
- **SDK Reference** – describes how to use the Luna HSM SDK to integrate your applications with 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.