THALES
Building a future we can all trust

Thales Cryptovisor K7 Cryptographic Module

LEVEL 3 NON-PROPRIETARY SECURITY POLICY

AS USED WITH LUNA CLOUD HSM OR HSM ON DEMAND SERVICES

Document Information

Document Part Number	002-010981-003
Release Date	November 12, 2023

Revision History

Revision	Date	Reason
Rev A	December 14, 2021	This SP is based on 002-010981-002, Rev A covering Cryptovisor firmware version 1.6. Initial release for firmware version 2.0.0.
Rev B	February 4, 2022	Updated mapping between services and roles in Table 5: Roles and Access Rights by Service.
Rev C	February 9, 2022	Added hardware part numbers 808-000048-003 and 808-000073-002.
Rev D	March 1, 2022	Corrected table of contents that was missing some sections.
Rev E	March 3, 2022	Corrected KAS claim in Table 8 to include options for use of X9.42 KDF with DH alongside SHA3 when being used with the Key Derivation service.
Rev F	April 25, 2022	Updated to address NIST SP800-56Ar3 transition questions.
Rev G	May 19, 2022	Update to address test laboratory comments.
Rev H	June 6, 2022	Update to address test laboratory comments.
Rev J	August 25, 2022	Updated to address CMVP comments during coordination. Added bootloader version 1.1.5.
Rev K	September 4, 2022	Update to address test laboratory comments.
Rev L	September 21, 2022	Minor typographical update to SHA-3 listing.
Rev M	February 10, 2022	Added firmware version 2.0.2.
Rev N	November 12, 2023	Added firmware version 2.0.5.

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
ADL	Authorized Device List
AES	Advanced Encryption Standard
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BBRAM	Battery Backed Random Access Memory
CEK	Content Encryption Key
CPV1	Cloning Protocol Version 1
CPV4	Cloning Protocol Version 4
CRC	Cyclic Redundancy Check
CSP	Critical Security Parameter
CVL	Component Validation List
CVM	CryptoVisor Manager
DAC	Device Authentication Certificate
DAK	Device Authentication Key
DEK	Domain Encryption master Key
DeMC	Device Messaging Certificate
DeMK	Device Messaging Key
DH	Diffie Hellman
DoMC	Domain Messaging Certificate
DoMK	Domain Messaging Keys
DOC	Domain Origin Certificate
DOK	Domain Origin Key
DRBG	Deterministic Random Bit Generator

Term	Definition
DSA	Digital Signature Algorithm
DSS	Digital Signature Standards
ECC	Elliptic Curve Cryptography
ECC DAC	Elliptic Curve Cryptography - Device Authentication Certificate
ECC DAK	Elliptic Curve Cryptography – Device Authentication Key
ECC HOC	Elliptic Curve Cryptography – Hardware Origin Certificate
ECC HOK	Elliptic Curve Cryptography – Hardware Origin Key
ECC MIC	Elliptic Curve Cryptography – Message Integrity Code
ECC MIK	Elliptic Curve Cryptography – Message Integrity Key
ECDH	Elliptic Curve Diffie Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EdDSA	Edwards-curve Digital Signature Algorithm
EFP	Environment Failure Protection
EMI	Electro-Magnetic Interference
EMC	Electro-Magnetic Compatibility
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
FSC	Firmware Signing Certificate
FSK	Firmware Signing Key
GCM	Galois Counter Mode
GSK	Global Storage Key
HOC	Hardware Origin Certificate
HOK	Hardware Origin Key
HSE-BBRAM	High Speed Erasable Battery Backed RAM.
HSM	Hardware Security Module / Host Security Module
HMAC	Hash-based Message Authentication Code

Term	Definition
ICD	Interface Control Document
I/O	Input / Output
IV	Initialization Vector
JOSE-CEK	JOSE – Content Encryption Key
JOSE-RK	JOSE – Response Key
JP(T)-CEK	Join Protocol (Target) – Content Encryption Key
JP(R)-CEK	Join Protocol (Root) – Content Encryption Key
JP-PTK	Join Protocol - PDO Transfer Key
JSON	JavaScript Object Notation
JOSE	JavaScript Object Signing and Encryption
JWE	JSON Web Encryption
KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-Base Key Derivation Function
KCV	Key Cloning Vector
KDF	Key Derivation Function
KEK	Key Encryption Key
KTS	Key Transport Scheme
LED	Light Emitting Diode
LSC	License Signing Certificate
LSK	License Signing Key
MAC	Message Authentication Code
MIC	Manufacturer’s Integrity Certificate
MIK	Manufacturer’s Integrity Key
MSK	Manufacturer’s Signature Key
NIST	National Institute of Science and Technology

Term	Definition
NDRNG	Non-Deterministic Random Number Generator
OAEP	Optimal Asymmetric Encryption Padding
ParEK	Partition Encryption Key
PCIe	Peripheral Component Interconnect
PCO	Partition Crypto Officer
PCU	Partition Crypto User
PDA	Provider Domain Administrator
PDO	Provider Domain Object
PEC	Password Encryption Certificate
PED	PIN Entry Device
PEK	Password Encryption Key
PFK	Partition Fragment Key
PKCS	Public-Key Cryptography Standards
POST	Power-On Self-Test
PSK	Partition Storage Key
PSO	Partition Security Officer
PU	Public User
RAM	Random Access Memory
RFC	Request For Comments
RNG	Random Number Generator
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SMK	Security Officer’s Master Key
SP	Special Publication
STM	Secure Transport Mode

Term	Definition
TUK3	Token Unwrapping Key 3
TWK3	Token Wrapping Key 3
USB	Universal Serial Bus
USK	User's Storage Key
XTC	MatriX Trusted Channel
XTC-epCK	XTC – ephemeral Client Key
XTC-PMK	XTC – Partition Messaging Key
XTC-PMc	XTC – Partition Messaging Certificate
XTC-PT	XTC – Partition Token
XTC-PToK.	XTC – Partition Token Key
XTC-PTuK	XTC – Partition Tunnel Key
XTC-SA	XTC – Secret AppID
XTC-TDK	XTC – Tunnel key Derivation Key
XTC-TTC	XTC – Tunnel Transport Key

REFERENCES

[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.

[FIPS 140-2] Federal Information Processing Standards Publication (FIPS PUB) 140-2, 'Security Requirements for Cryptographic Modules', May 25, 2001 (including change notices 12-02-2002).

[FIPS 140-2 IG] NIST, Implementation Guidance for FIPS 140-2 and the Cryptographic Module Validation Program, May 1, 2021.

[FIPS 180-4] Federal Information Processing Standards Publication 180-4, Secure Hash Standard (SHS), NIST, August 2015.

[FIPS 186-4] Federal Information Processing Standards Publication 186-4, Digital Signature Standards (DSS), NIST, July 2013.

[FIPS 197] Federal Information Processing Standards Publication 197, Specification for the Advanced Encryption Standard (AES), November 26, 2001.

[FIPS 202] Federal Information Processing Standards Publication 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August 2015.

[FIPS 198-1] Federal Information Processing Standards Publication 198-1, The Keyed-Hash Message Authentication Code (HMAC), July 2008.

[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.

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

[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 7516] RFC 7516, 'JSON Web Encryption (JWE)', M. Jones, May 2015.

[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 – SEC 2: 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, Morris Dworkin, December 2001.

[SP800-38B] NIST Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication, May 2005 (with October 2016 updates).

[SP800-38D] NIST Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.

[SP800-38E] NIST Special Publication 800-38E, Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices, January 2010.

[SP800-38F] NIST Special Publication 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, December 2012.

[SP800-56Ar3] NIST Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, Revision 3, April 2018.

[SP800-56Br2] NIST Special Publication 800-56B, Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography, Revision 2, March 2019.

[SP800-56Cr2] NIST Special Publication 800-56C, Recommendation for Key-Derivation Methods in Key-Establishment Schemes, Revision 1, April 2018.

[SP800-67r2] NIST Special Publication 800-67, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Rev 1, January 2012.

[SP800-90Ar1] NIST Special Publication SP800-90A, Recommendation for Random Number Generation Using Deterministic Bit Generators, Rev1, June 2015.

[SP800-90B] NIST, SP800-90B, “Recommendation for the Entropy Sources Used for Random Bit Generation”, Version 1.0, January 2018.

[SP800-108] NIST Special Publication SP800-108, Recommendation for Key Derivation Using Pseudorandom Functions (Revised), October 2009.

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

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

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

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

PREFACE

This document deals only with operations and capabilities of the Thales Cryptovisor K7 Cryptographic Module in the technical terms of [FIPS 140-2], 'Security Requirements for Cryptographic Modules', 12-03-2002.

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.
- 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 Introduction

1.1 Purpose

This document describes the security policies enforced by Thales Cryptovisor K7 Cryptographic Module.

1.2 Scope

This document applies to hardware versions 808-000048-002, 808-000048-003, 808-000073-001 and 808-000073-002 with firmware version 2.0.0, 2.0.2 and 2.0.5 and bootloader versions 1.1.1, 1.1.2, 1.1.4 and 1.1.5 and where:

- 808-000048-002 and 808-000048-003 correspond to a module with fans on the outside of the metal enclosure factory installed;
- 808-000073-001 and 808-000073-002 correspond to a module with extra heatsinks installed (instead of fans as pictured);
- 808-000048-002 and 808-000048-003 are functionally equivalent with the difference being limited to the supply choice for one of the non-security enforcing internal components; and
- 808-000073-001 and 808-000073-002 are functionally equivalent with the difference being limited to the supply choice for one of the non-security enforcing internal components.

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

The Thales Cryptovisor K7 Cryptographic Module is used as the HSM embedded in the Thales Cloud HSM services, HSM on Demand.

1.3 Validation Overview

The cryptographic module meets all Level 3 requirements for FIPS 140-2 as summarized in the table below:

Table 1: FIPS 140-2 Security Levels

Security Requirements Section	Level
Cryptographic Module Specification	3
Cryptographic Module Ports and Interfaces	3
Roles and Services and Authentication	3
Finite State Machine Model	3
Physical Security	3 + EFP
Operational Environment	N/A
Cryptographic Key Management	3

Security Requirements Section	Level
EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	3
Cryptographic Module Security Policy	3

1.4 Functional Overview

The Thales Cryptovisor K7 Cryptographic Module is a multi-chip embedded cryptographic module in the form of a PCIe card that typically resides within a custom computing or secure communications appliance. The cryptographic module is contained in its own secure enclosure that provides physical resistance to tampering. The cryptographic boundary of the module is defined to encompass all components inside the secure enclosure on the PCIe card. Figure 1, 'Thales Cryptovisor K7 Cryptographic Module cryptographic boundary' in section 2.1, 'Module Specification' depicts the Thales Cryptovisor K7 Cryptographic Module. Figure 2, 'Thales Luna PED and Provider Domain Administrator (PDA) iKey' in section 2.3, 'Trusted Path' depicts the Thales Luna PIN Entry Device (PED) and iKey used for authentication of the Provide Domain Administrator role.

The module may be explicitly configured to operate in either FIPS 140-2 Approved mode or in a non-FIPS mode of operation. Note that selection of operating in FIPS 140-2 approved mode occurs at initialization of the cryptographic module and cannot be changed during normal operation without zeroizing the module's non-volatile memory. Section 3.2, 'Approved Mode of Operation' provides additional information for configuring the module in FIPS 140-2 approved mode of operation.

The module is accessed directly (i.e., electrically) over the PCIe communications interface. The Thales Luna PED can be connected to the module's USB port for authentication.

The module provides secure key generation and storage for symmetric keys and asymmetric key pairs along with symmetric and asymmetric 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 Luna ICD command interface.

The module may host multiple "user partitions" that are cryptographically separated and are presented as "virtual tokens" to user applications. A single "admin partition" exists that is dedicated to the Provider Domain Administrator (PDA) role. Each partition must be separately authenticated in order to make it available for use.

2 Module Overview

2.1 Module Specification

The cryptographic module is a multi-chip embedded hardware module.

The cryptographic boundary of the module is shown below. The cryptographic boundary is defined as the metal enclosure on the top and bottom sides of the PCIe card as outlined. The fans depicted alongside the removable backup battery are not included in the cryptographic boundary.

Figure 1: Thales Cryptovisor K7 Cryptographic Module cryptographic boundary

2.2 Ports and Interfaces

The module supports the following physical ports and interfaces:

- PCIe interface;
- USB port;
- Serial port;
- Power supply;
- Battery;
- LED;
- External event input; and
- Decommission input.

Table 2: Mapping of FIPS 140-2 Interfaces to Physical and Logical Interfaces

FIPS 140-2 Interface	Physical Interface	Logical Interface
Data Input	PCIe interface	Data I/O Luna ICD XTC channel for ICD Domain Administration API Bootloader command protocol
	USB	Physical Trusted Path (Thales Luna PED)
	Serial interface	Bootloader command protocol
Data Output	PCIe interface	Data I/O Luna ICD XTC channel for ICD Domain Administration API Bootloader command protocol
	USB	Physical Trusted Path (Thales Luna PED)
	Serial Port	Bootloader command protocol
Control Input	PCIe interface	Data I/O Luna ICD XTC channel for ICD Domain Administration API
	External event jumper	N/A
	Decommission jumper	N/A
	Serial Port	Cryptovisor Communication Path
Status Output	PCIe interface	Data I/O Luna ICD XTC channel for ICD Domain Administration API Bootloader command protocol
	USB	Physical Trusted Path (Thales Luna PED)
	LED	N/A
	Serial Port	Bootloader command protocol

FIPS 140-2 Interface	Physical Interface	Logical Interface
Power	5V and 1.8V (generated from 12V power supply via PCIe interface)	N/A
	3.6V battery	N/A

2.3 Trusted Path

The module uses a Thales Luna PED as an external data input/output device in support of authenticating the PDA role. The Thales Luna PED connects to the module's USB port and is used to pass authentication data to and from the module via a physical trusted path. Authentication data that is output to the Thales Luna PED is stored in an iKey USB device connected to the Thales Luna PED.

Any iKey, once data has been written to it, is an Identification and Authentication device and must be safeguarded accordingly by the administrative or operations staff responsible for the operation of the module within its deployed environment.

The following types of iKey are used with the Thales Luna PED:

> Blue (HSM) iKey – for the storage of PDA authentication data[1].

Figure 2: Thales Luna PED and Provider Domain Administrator (PDA) iKey

[1]Separate iKey are used when M of N token splitting is used to share responsibilities for this role between different operators.

2.4 Secure Messaging

The module implements two secure messaging interfaces:

- Each user partition uses a secure messaging feature called XTC. An XTC channel is a cryptographic tunnel established between a partition and the client running on a host. The XTC channel is designed to provide authenticity, confidentiality and integrity of all Luna ICD commands transmitted over it.

  XTC establishes a shared secret between the cryptographic module and client using a [SP800-56Ar3] compliant key agreement scheme as covered in section 'Key Import and Export' on page 68. Traffic is encrypted using AES-GCM and using a 256-bit key.

- The Domain Administration API is implemented using JWE [RFC 7516] where RSA-OAEP (using modulus length 4096-bits) is used to encrypt a 256-bit content encryption key that itself is used to encrypt message payloads using AES-GCM[2].

2.5 Roles and Services

2.5.1 Roles

The Thales Cryptovisor K7 Cryptographic Module supports the following roles:

Table 3: Thales Cryptovisor K7 Cryptographic Module Roles

Role	Responsibilities
Provider Domain Administrator (PDA)	> Domain-level role. > Creates Partition Domain Object (PDO) containing the Authorized Device List (ADL). > Registers HSM by serial number to a Domain's ADL. > Activates / Deactivates a Domain on an HSM. > Configures HSM level policies. > Updates module firmware for initialized modules.[3] > Initialize the CVM Role.
Cryptovisor Manager (CVM)	> Domain-level role. > Creates partitions. > Imports partition and session containers to a HSM.
Partition Security Officer (PSO)	> User partition-level role. > Configures container policies for user partition. > Initialize the PCO Role.

[2] The domain administration command `get-device-certs` command is allowed to execute unencrypted for bootstrapping purposes as it returns the first public key used to access all others.

[3] Any firmware loaded into this module that is not shown on the module certificate is out of the scope of this validation and requires a separate FIPS 140-2 validation.

Role	Responsibilities
Partition Crypto Officer (PCO)	> User partition-level role. > Generates partition cryptographic keys for use by cryptographic services accessing the partition. > Uses container keys[4] in order to support cryptographic services. > Initialize the PCU Role.
Partition Crypto User (PCU)	> User partition-level role. > Uses partition cryptographic keys.
Public User (PU)	> Zeroizes HSM from local interfaces via command (i.e. not permissible over XTC). > Retrieval of status information. > Collects module utilization statistics. > Power cycle.

The mapping of the cryptographic module's roles to the roles defined in FIPS 140-2 can be found in the table below.

Table 4: Mapping of FIPS 140-2 Roles to Module Roles

FIPS 140-2 Role	Cryptovisor HSM Role	Role Scope
Crypto Officer	Provider Domain Administrator	Module
	Cryptovisor Manager	Module
	Partition Security Officer	User Partition
User	Crypto Officer	User Partition
	Crypto User	User Partition
Unauthenticated User	Public User	Module/Partition

2.5.2 Services

All services listed in the table below can be accessed in FIPS 140-2 Approved mode and use the security functions listed in Table 8: FIPS-Approved Algorithm Implementation in section 2.9.1, 'Approved Algorithms'.

[4] Asymmetric Key Pairs (general partition or session keys) and Symmetric Keys (general partition or session keys)

When the module is operating in FIPS 140-2 Approved mode as described in section 3.2, ‘Approved Mode of Operation’, the non-Approved Security Functions in section 2.9.2, ‘Non-Approved Algorithm Implementations’ are disabled and cannot be used for these services.

The non-Approved functions can only be accessed through the services when the module is in non-Approved mode.

For a complete description of CSP referenced from the table, please see section 2.10, ‘Critical Security Parameters’.

Note: In the ‘Type(s) of Access’ column:

- where a CSP is listed as ‘Write’ the module service is responsible for storing the target CSP for future use; and
- where a CSP is listed as ‘Use’ the target service will either retrieve from storage or generate the target CSP ahead of using it.

Table 5: Roles and Access Rights by Service

Service	Critical Security Parameters	Type(s) of Access (Write/Use)	Role					
			PDA	CVM	PSO	PCO	PCU	PU
Show Status	N/A	N/A	x	x	x	x	x	x
Self-test	N/A	N/A	x	x	x	x	x	x
Get Device Certs	DeMC, HOC, MIC, ROOT	Use	x	x	x	x	x	x
Receive/Process JWE/JOSE Message for Device	JOSE-RK, JOSE-CEK, DeMK	Use	x	x	x	x	x	x
Receive/Process JWE/JOSE Message for Domain	JOSE-RK, JOSE-CEK, DoMK	Use	x	x	x	x	x	x
Create Domain	DRBG Key, DRBG V.	Use	-	-	-	-	-	x
	DRBG Key, DRBG V, PDA Pin (as written to iKey on PDA role creation), SMK, DOK, DOC, DoMK, DoMC, DRBG Key, DRBG V	Write						
Modify Domain Authorized Device List	PDA Pin (as read from iKey), SMK	Use	x	-	-	-	-	-
Set Domain Policy	PDA Pin (as read from iKey), SMK	Use	x	-	-	-	-	-

Service	Critical Security Parameters	Type(s) of Access (Write/Use)	Role					
			PDA	CVM	PSO	PCO	PCU	PU
Download Provider Domain Object (Session Device)	ROOT, HOC, MIC, ECC MIC, ECC HOC, PDA Pin, SMK, JP-PTK, JP(R)-CEK, DoMK	Use	x	-	-	-	-	-
	JP(T)-CEK	Write						
Transfer Provider Domain Object (Root Device)	DRBG Key, DRBG V, ROOT, HOC, MIC, ECC MIC, ECC HOC, JP(T)-CEK, JP-PTK, DoMC	Use	x	-	-	-	-	-
	DRBG Key, DRBG V	Write						
Activate a device	PDA Pin	Use	x	-	-	-	-	-
Deactivate a Device	N/A	N/A	x	-	-	-	-	-
Login	Authentication data, DRBG Key, DRBG V, PEC/PEK	Use	x	x	x	x	x	-
	DRBG Key, DRBG V, USK, PSK	Write						
Logout	N/A	N/A	x	x	x	x	x	-
Retrieve Domain Certs	DOC, DMC, HOC, MIC	Use	x	x	x	x	x	-
Get Domain Info	N/A	N/A	x	x	x	x	x	x
Create Partition	CVM Password, DOK, DRBG Key, DRBG V	Use	-	x	-	-	-	-
	DRBG Key, DRBG V, PFK, PMK, PSK, USK, XTC- XTC-PMC, XTC-PToK	Write						
Delete Partition	CVM Password	Use	-	x	-	-	-	-
Initialize Partition	USK	Use	-	-	x	-	-	-
Configure Partition Policy	N/A	N/A	-	-	x	-	-	-

Service	Critical Security Parameters	Type(s) of Access (Write/Use)	Role					
			PDA	CVM	PSO	PCO	PCU	PU
Open XTC Session	DRBG Key, DRBG V, XTC-epCK, XTC-PMK, XTC-PMC, XTC-PT, XTC-TDK, XTC-TTC	Use	-	-	x	x	x	x
	DRBG Key, DRBG V, XTC-PToK, XTC-PTuK, XTC-PT, XTC-SA	Write						
Process Incoming/Outgoing XTC Traffic	XTC-SA, XTC-PToK, XTC-PTuK, XTC-PT	Use	-	-	x	x	x	x
Create Partition User	PSO Admin Password (for CO) or CO Password (for CU), PSK, USK	Use	-	-	x	x	-	-
	CO or CU Password.	Write						
Re-initialize Partition User	PSO Admin Password, PSK, USK	Use	-	-	-	x	-	-
Retrieve Partition Certs	HOC, MIC, DOC, XTC-PMC	Use	x	x	x	x	x	x
Partition Export	DEK, ParEK, PFK	Use	x	x	x	x	x	x
Partition Import	CVM Password, DEK, ParEK, PFK	Use	-	x	-	-	-	-
	XTC-PMK, XTC-PMC, USK, PSK, Partition symmetric and Asymmetric key objects as contained in partition	Write						
Session Object Export	DEK, ParEK, PFK	Use	x	x	x	x	x	x
Session Object Import	CVM Password, DEK, ParEK, PFK	Use	-	x	-	-	-	-
Firmware Update	FSC, ROOT	Use	x	-	-	-	-	-
License Update	LSC, ROOT	Use	x	-	-	-	-	-

Service	Critical Security Parameters	Type(s) of Access (Write/Use)	Role					
			PDA	CVM	PSO	PCO	PCU	PU
Key Generation	DRBG Key, DRBG V, USK[5]	Use	-	-	-	x	-	-
	DRBG Key, DRBG V, Symmetric keys (general partition or session keys)	Write						
Key Pair Generation	DRBG Key, DRBG V, USK[6]	Use	-	-	-	x	-	-
	DRBG Key, DRBG V, Asymmetric key pairs (general partition or session keys)	Write						
Wrap Symmetric Key	USK[7], DRBG Key, DRBG V, KTS symmetric/asymmetric wrapping key	Use	-	-	-	x	-	-
	DRBG Key, DRBG V	Write						
Wrap Asymmetric Key	USK[8], DRBG Key, DRBG V, KTS symmetric wrapping key	Use	-	-	-	x	-	-
	DRBG Key, DRBG V.	Write						
Unwrap Symmetric/Asymmetric Key	KTS symmetric unwrapping key, symmetric unwrapping key	Use	-	-	-	x	-	-
	Symmetric or Asymmetric unwrapped key	Write						

[5] If keys are stored long-term on the module as a ‘token’ object (rather than being a ‘session’ object that will automatically be zeroized on session closure).
[6] If keys are stored long-term on the module as a ‘token’ object (rather than being a ‘session’ object that will automatically be zeroized on session closure).
[7] If either the wrapping key or key to be wrapped is a ‘token’ object rather than being a ‘session’ object.
[8] If either the wrapping key or key to be wrapped is a ‘token’ object rather than being a ‘session’ object.

Service	Critical Security Parameters	Type(s) of Access (Write/Use)	Role					
			PDA	CVM	PSO	PCO	PCU	PU
Clone Partition Object (CPV1)	DRBG Key, DRBG V, ROOT, MIC, HOC and TUK3, TWK3, KEV_s, KEV_t, KCV and Cloning Transfer Key	Use	-	-	-	x	x	-
	DRBG Key, DRBG V	Write						
Open Cloning Session / Clone Partition Object (CPV4)	USK, DRBG Key, DRBG V, CPV4 Cookie Key Derivation Key, CPV4 Cookie Key, CPV4 Key Agreement Private Key, CPV4 Key Agreement Public Key, CPV4 Key Agreement Shared Secret, KCV, CPV4 Key Transport Key, CPV4 Session Key, EC Attestation Key	Use	-	-	-	x	x	-
	DRBG Key, DRBG V, CPV4 Session Key	Write	-	-	-	x	x	-
Encrypt/Decrypt (Symmetric Algorithm)	USK[9], DRBG Key, DRBG V, Symmetric keys (general partition or session keys)	Use	-	-	-	x	x	-
	DRBG Key, DRBG V	Write						
Encrypt/Decrypt (Asymmetric Algorithm)	USK[10], DRBG Key, DRBG V, Asymmetric keys (general partition or session keys)	Use	-	-	-	x	x	-
	DRBG Key, DRBG V	Write						

[9] Where symmetric keys used are “token” rather than “session” objects.
[10] Where asymmetric private key used is a “token” rather than “session” object.

Service	Critical Security Parameters	Type(s) of Access (Write/Use)	Role					
			PDA	CVM	PSO	PCO	PCU	PU
Signature Generation (Public Key Cryptography)	USK[11], DRBG Key, DRBG V, RSA, DSA, ECDSA private keys (general partition or session keys)	Use	-	-	-	x	x	-
	DRBG Key, DRBG V	Write						
Signature Verification (Public Key Cryptography)[12]	RSA, DSA, ECDSA public keys (general partition or session keys)	Use	-	-	x	x	x	-
Generate Hash Value	N/A	N/A	-	-	x	x	x	-
MAC Generation	USK[13], Symmetric keys (general partition or session keys)	Use	-	-	-	x	x	-
MAC Verification	USK[14], Symmetric keys (general partition or session keys)	Use	-	-	-	x	x	-
Key Derivation	USK[15], DRBG Key, DRBG V, Symmetric keys (general partition or session keys), Asymmetric keys (general partition or session keys)	Use	-	-	-	x	-	-
	DRBG Key, DRBG V, Symmetric keys (general partition or session keys)	Write						
Retrieve DRBG output for export.	DRBG Key, DRBG V	Use	-	-	x	x	x	-
	DRBG Key, DRBG V	Write						
Store Data Object[16]	Non-cryptographic data	Write	-	-	x	x	x	x

[11] Where symmetric or private keys used are “token” rather than “session” objects.
[12] The PSO is able to validate signatures using public keys where **CKA_PRIVATE** is set to **0**.
[13] Where symmetric keys used are “token” rather than “session” objects.
[14] Where symmetric keys used are “token” rather than “session” objects.
[15] Where symmetric or private keys used are “token” rather than “session” objects.
[16] PSO and PU are exclusively permitted to create data objects where **CKA_PRIVATE** is set to **0**.

Service	Critical Security Parameters	Type(s) of Access (Write/Use)	Role					
			PDA	CVM	PSO	PCO	PCU	PU
Read Data Object[17]	Non-cryptographic data	Use	-	-	x	x	x	x
Export Audit Log Data	Non-cryptographic data	Use	x	x	-	-	-	x
Reset Card following Tamper Event (not zeroized)	N/A	N/A	x	-	-	-	-	-
Set Device Time	N/A	N/A	-	x	-	-	-	-
Zeroize Device	N/A	N/A	x	x	x	x	x	x
Request authentication and execution of main firmware	FSC, ROOT	Use	x	x	x	x	x	x

2.6 Authentication

All roles except for the Public User must authenticate to the module by providing their authentication data. Table 5: Roles and Access Rights by Service and Table 6: Roles and Required Identification and Authentication explain the type and strength of the authentication data supported for each role.

All roles must authenticate using either presentation of an iKey or a password. For the PSO, PCO and PCU roles, when the role is initialized, the operator enters the initial password for the role. The password is delivered to the module encrypted with public key from the module's Password Encryption Certificate (PEC) using RSA-OAEP and a random nonce to prevent replay attacks.

For the CVM role, the password is generated by the HSM and returned AES-256 encrypted in GCM mode under a JOSE Content Encryption Key (CEK) supplied to the module by the PDA. When re-submitted to the module during authentication, the password is RSA-OAEP encrypted under the public key from the Domain Messaging Certificate.

For the PDA, authentication data is written to the PED Key over a Trusted Path (direct physical connection) to the Thales Luna PED.

Table 6: Roles and Required Identification and Authentication

Role	Type of Authentication	Authentication Data
Provider Domain Administrator	Identity-based	Authentication token (iKey) with optional PIN.
Cryptovisor Manager	Identity-based	Password

[17] PSO and PU are exclusively permitted to read data objects where **CKA_PRIVATE** is set to **0**.

Role	Type of Authentication	Authentication Data
Partition Security Officer	Identity-based	Password
Partition Crypto Officer	Identity-based	Password
Partition Crypto User	Identity-based	Password

Table 7: Strengths of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
iKey (PDA)	Authentication is based on presentation of a 48-byte random value generated when a role is initialized and stored on an iKey. The probability of guessing the authentication data in a single attempt is 1 in 2^{384}. With a maximum of 6000[18] failed login attempts per minute, the thresholds required by FIPS 140-2 can never be reached.
Password (CVM)	Authentication is based on presentation of a 16-byte random secret generated by the cryptographic module and output to the user as a base64 encoded ASCII string. The probability of guessing the authentication data in a single attempt is 1 in 2^{128}. With a maximum of 6000[18] failed login attempts per minute, the thresholds required by [FIPS 140-2] can never be reached.
Password (PSO, PCO, PCU).	Authentication is based on presentation of a user provided byte array (minimum 7 bytes). The probability of guessing the secret in a single attempt is 1 in 2^{56}. With a maximum of 8000[18] failed login attempts per minute, the thresholds required by FIPS 140-2 can never be reached.

2.6.1 M of N

The cryptographic module supports the use of an M of N (up to N=16) secret sharing authentication scheme for the PDA role. M of N authentication provides the capability to enforce multi-person control over the functions associated with this role.

The M of N capability is based on Shamir's threshold scheme. The cryptographic module splits the randomly-generated authentication data into "N" pieces, known as splits, and stores each split on an iKey. Any "M" of these "N" splits must be transmitted to the cryptographic module by inserting the corresponding iKeys into the Thales Luna PED in order to reconstruct the original secret.

2.7 Physical Security

The Thales Cryptovisor K7 Cryptographic Module is a multi-chip embedded module as defined by [FIPS 140-2], section 4.5. The module is enclosed in a strong metal enclosure that provides tamper-evidence. Any tampering that might compromise a module's security is detectable by visual inspection of the physical integrity of a module. The PDA should perform a visual inspection of the module at regular intervals.

[18] This is based on testing and the bandwidth limitation of authentication (e.g., 100 attempts per second, or 6000 attempts per minute). So, it represents an actual result of testing.

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

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

2.7.1 External Event

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

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

2.7.2 PCIe Card Removal

The module detects removal from the PCIe slot in both the powered-on state and the powered-off state. If the card is removed from the PCIe slot, the event is logged.

2.7.3 EFP

The module 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 monitored in both the powered-on state and the powered-off state.

In the event that the module senses an out-of-range temperature or over voltage, the module will reset itself, erase the Token Module Variable Key, clear all working memory and log the event. The module can be reset and placed back into operation when proper operating conditions have been restored.

Under-voltage conditions cannot be distinguished from a power cycle.

In the event that the module senses an under voltage, the module will reset itself and clear all working memory. The Login Token Encryption Key will not be erased. The module can be reset and placed back into operation when proper operating conditions have been restored.

2.7.4 Decommission

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

In the event of a decommission signal, the module will erase the Key Encryption Key (KEK), reset itself, clear all working memory and log the event.

This provides the capability to prevent access to sensitive objects in the event that the module has become unresponsive or has lost access to primary power.

The module can be reset, re-initialized and placed back into operation when the decommission signal is removed.

2.7.5 Secure Transport Mode

Secure Transport Mode (STM) is a feature that allows the integrity of the module to be verified when the module is shipped from one location to another or placed in storage.

When a module is placed into STM, a random string and a fingerprint of the internal state of the module is output from the module. The fingerprint is a SHA2-256 digest of the random string, a randomly generated nonce, module CSPs, firmware, module configuration information and non-volatile memory. The nonce is stored in the HSE-BBRAM that is erased in response to an External Event, Decommission signal and EFP violations.

While in STM, the module is in a reduced mode of operation, which only allows the module to be taken out of STM. If the module has been initialized, only the PDA can put the modules into STM and take it out of STM. If the HSM is in a zeroized state, only the public user can put the module into STM and take it out of STM.

The module can be taken out of STM by entering the random user string. The module will recalculate and output the fingerprint. It is the operator's responsibility to verify that the fingerprint output matches the fingerprint initially output when the module was put in to STM.

2.7.6 Fault Tolerance

If power is lost to a module for whatever reason, the module shall, at a minimum, maintain itself in a state that it can be placed back into operation when power is restored without compromise of its functionality or permanently stored data.

A module shall maintain its secure state[19] in the event of data input / output failures. When data input / output capability is restored the module will resume operation in the state it was prior to the input / output failure.

2.8 Operational Environment

The module uses a non-modifiable operational environment. The requirements for a modifiable operating environment do not apply.

2.9 Cryptographic Key Management

2.9.1 Approved Algorithms

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

> **SafeNet Accelerated Cryptographic Library v1.0**

- CAVP certificate page for library – AES, DSA, RSA, SHA, HMAC, ECDSA, RSADP, DRBG, KAS-ECC-SSC, KAS-FFC-SSC, KDA, CVL (X9.42, X9.63).

> **SafeNet Cryptographic Library v1.0**

- CAVP certificate page for library – RSADP, RSA.

> **K7 Boot Loader v1.1.1**

- CAVP certificate for library – RSA and SHA2.

> **K7 Bootloader v1.1.2**

- CAVP certificate for library – RSA and SHA2.

[19] A secure state is one in which either the cryptographic module is operational and its security policy enforcement is functioning correctly, or it is not operational and all sensitive material is stored in a cryptographically protected form.

NOTE The module supports two additional versions of the K7 bootloader, 1.1.4 and 1.1.5. Changes made between versions 1.1.2, 1.1.4 and 1.1.5 do not impact the internal cryptographic libraries used by the module and re-testing for versions 1.1.4 and 1.1.5 was not required based on guidance provided in [FIPS 140-2 IG], 1.4, 'Binding of Cryptographic Algorithm Validation Certificates'.

NOTE The following libraries referenced above have redundant certificate listings not used by the current firmware version of the cryptographic module:

> **SafeNet Accelerated Cryptographic Library v1.0**

- Cert KAS 195 includes a listing for KAS-ECC and KAS-FFC.
- CVL Cert #2047 ECDSA SigGen for B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521.
- CVL Cert #2043 RSA Decryption Primitive.

> **SafeNet Cryptographic Library v1.0**

- Cert KAS 196 includes a listing for KAS-FFC;
- Cert AES 5653 includes a listing for GCM and GMAC;
- Cert ECDSA 1527 includes a key Gen listing for B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521;
- Cert SHS cert #4534 for: SHA2-256 and SHA2-512;
- Cert DSA #1453 including DSA KeyGen and PQGGen; and
- Cert #A1170, KAS-ECC-SSC, KAS-FFC-SSC and PBKDF components.
- CVL Cert #2044 RSA Decryption Primitive.

The approved algorithms implemented by the module alongside their mapping to the certificates above alongside algorithms use by service are listed in the Table 8 below.

Listings in the 'Use / Function' column map to services listed in Table 5.

Table 8: FIPS-Approved Algorithm Implementation

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Symmetric Encryption/Decryption				
AES #5652.	**Algorithm:** AES. **Standards:** [FIPS 197], [SP800-38A], [SP800-38D], [SP800-38E] and [SP800-38F].	**Mode:** CBC, CFB128, CFB8, CTR, ECB, GCM[20], KW, KWP, OFB, XTS[21].	**Key size:** 128 and 256-bits - all modes. 192-bits – CBC, CFB128, CFB8, CTR, ECB, GCM, KW and KWP only.	Used to support the following services: Activate a device, Clone Partition Object (CPV1), Open Cloning Session / Clone Partition Object (CPV4), Create Domain, Create Partition, Create Partition User, Download Provider Domain Object, Encrypt/Decrypt (Symmetric Algorithm), Initialize Partition, Key Derivation, Key Generation, Key Pair Generation, Login, Modify Domain Authorized Device List, Open XTC Session, Partition Export, Partition Import, Receive/Process JWE/JOSE Message for Device, Receive/Process JWE/JOSE Message for Domain, Process Incoming/Outgoing XTC Traffic, Re-initialize Partition User, Self-test, Session Object Export, Session Object Import, Signature Generation (Public Key Cryptography), Signature Verification (Public Key Cryptography), Transfer Provider Domain Object, Unwrap Symmetric/Asymmetric Key, Wrap Asymmetric Key and Wrap Symmetric Key.

[20] The module generates IVs internally using the approved DRBG where all IV used are 128-bits in length.
[21] XTS-AES is only supported for consumption by users with the 'Encrypt/Decrypt (Symmetric Algorithm)' service. When used, output from this service should be used exclusively for data storage.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Hashing				
SHS #4533.	**Algorithm:** SHA. **Standards:** [FIPS 186-4].	**Methods:** SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 (Byte Only).	N/A.	Used to support the following services: Activate a device, Clone Partition Object (CPV1), Open Cloning Session / Clone Partition Object (CPV4), Create Domain, Create Partition, Create Partition User, Download Provider Domain Object, Encrypt/Decrypt (Symmetric Algorithm),Encrypt/Decrypt (Asymmetric Algorithm), Firmware Update, Generate Hash Value, Initialize Partition, Key Derivation, Key Generation, Key Pair Generation, Login, MAC Generation, MAC Verification, Modify Domain Authorized Device List, Open XTC Session, Partition Export, Partition Import, Receive/Process JWE/JOSE Message for Device, Receive/Process JWE/JOSE Message for Domain, Process Incoming/Outgoing XTC Traffic, Re-initialize Partition User, Self-test, Session Object Export, Session Object Import, Signature Generation (Public Key Cryptography), Signature Verification (Public Key Cryptography), Transfer Provider Domain Object, Unwrap Symmetric/Asymmetric Key, Wrap Asymmetric Key and Wrap Symmetric Key.
#A1171.	**Algorithm:** SHA3. **Standard:** [FIPS 202].	**Methods:** SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE-128, SHAKE-256 (Byte Only).	N/A.	Used to support the following services: Generate Hash Value, Key Derivation, Key Generation, Key Pair Generation, Login, MAC Generation, MAC Verification, Self-test, Signature Generation (Public Key Cryptography), Signature Verification (Public Key Cryptography), Unwrap Symmetric/Asymmetric Key, Wrap Asymmetric Key and Wrap Symmetric Key.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
SHS #3951.	**Algorithm:** SHA. **Standards:** [FIPS 186-4].	**Methods:** SHA1, SHA2-384 (Byte Only).	N/A.	Used to support the following services when using bootloader 1.1.1: Request authentication and execution of main firmware.
SHS #3952.	**Algorithm:** SHA. **Standards:** [FIPS 186-4].	**Methods:** SHA1, SHA2-384 (Byte Only).	N/A.	Used to support the following services when using bootloaders 1.1.2, 1.1.4 and 1.1.5: Request authentication and execution of main firmware.
Message Authentication Code				
HMAC #3766.	**Algorithm:** HMAC. **Standard:** [FIPS 198-1].	**Methods:** HMAC-SHA1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512.	**Mac size:** 10-to-64 bytes (dependent on hash). **Key size:** key size < block size, key size = block size, key size > block size.	Used to support the following services: Key Derivation, MAC Generation and MAC Verification, Self-test.
HMAC #A1171.	**Algorithm:** HMAC. **Standard:** [FIPS 198-1].	**Methods:** HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512.	**Mac size:** 112-to-512 bits increment 8 (dependent on hash). **Key size:** 192-to-1152 bits increment 8 bits.	Used to support the following services: Key Derivation, MAC Generation and MAC Verification, Self-test.
AES #5652.	**Algorithm:** AES. **Standard:** [FIPS 197] and [SP800-38B].	**Methods:** CMAC.	**Key size:** 128, 192, 256-bits.	Used to support the following services: Key Derivation, MAC Generation and MAC Verification, Self-test.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Asymmetric				
RSA #3042.	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Method:** Key Generation, Signature Generation, Signature Verification. **Signature Type:** ANSI X9.31, PKCS #1-v1.5, PKCS-PSS. **Hash options:** Signature Generation (PKCS #1-v1.5 and PKCS-PSS): SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Generation (ANSI X9.31): SHA2-224, SHA2-256, SHA2-384, SHA2-512. Signature Verification (PKCS #1-v1.5 and PKCS-PSS): SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification (ANSI X9.31): SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512. Vendor affirmed using [FIPS 140-2 IG], A.11, 'The Use and the Testing Requirements for the Family of Functions defined in FIPS 202', when using SHA3.	**Modulus length:** 2048, 3072 – Key Generation, Signature Generation and Signature Verification. 1024 – Signature Verification only.	Used to support the following services: Self-test and Signature Verification (Public Key Cryptography). This implementation is used with the below services when **Partition Policy (16) Operate without RSA blinding** is enabled: Signature Generation (Public Key Cryptography) and Key Pair Generation. This policy is enabled by default for all modules.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A1171.	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Method:** Key Generation, Signature Generation, Signature Verification. **Signature Type:** ANSI X9.31, PKCS #1-v1.5, PKCS-PSS. **Hash options:** Signature Generation (PKCS #1-v1.5 and PKCS-PSS): SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Generation (ANSI X9.31): SHA2-224, SHA2-256, SHA2-384, SHA2-512. Signature Verification (PKCS #1-v1.5 and PKCS-PSS): SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification (ANSI X9.31): SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512. Vendor affirmed using [FIPS 140-2 IG], A.11, 'The Use and the Testing Requirements for the Family of Functions defined in FIPS 202', when using SHA3.	**Modulus length:** 4096 – Key Generation, Signature Generation and Signature Verification. **Vendor Note:** Key sizes up to modulus length 8192-bit are supported for key generation, Signature Generation and Signature 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.	Used to support the following services: Clone Partition Object (CPV1), Open Cloning Session / Clone Partition Object (CPV4), Create Domain, Create Partition, Download Provider Domain Object, Firmware Update, License Update, Open XTC Session, Receive/Process JWE/JOSE Message for Device, Receive/Process JWE/JOSE Message for Domain and Self-test, Signature Verification (Public Key Cryptography). This implementation is used with the below services when **Partition Policy (16) Operate without RSA blinding** is enabled: Signature Generation (Public Key Cryptography) and Key Pair Generation. This policy is enabled by default for all modules.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
RSA #3043.	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Method:** Key Generation. **Key Generation Mode:** B.3.3 and B.3.6.	**Modulus length:** 2048, 3072 – Key Generation.	Used to support the following services: Key Pair Generation. This implementation is used when **Partition Policy (16) Operate without RSA blinding** is disabled. This policy is enabled by default for all modules.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A1170.	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Method:** Key Generation. **Signature Type:** ANSI X9.31, PKCS #1-v1.5, PKCS-PSS. **Hash options:** Signature Generation (PKCS #1-v1.5 and PKCS-PSS): SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Generation (ANSI X9.31): SHA2-224, SHA2-256, SHA2-384, SHA2-512. Signature Verification (PKCS #1-v1.5 and PKCS-PSS): SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification (ANSI X9.31): SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512. **Key Generation Mode:** B.3.3 and B.3.6 Vendor affirmed using [FIPS 140-2 IG], A.11, 'The Use and the Testing Requirements for the Family of Functions defined in FIPS 202', when using SHA3.	**Modulus length:** 4096 – Key Generation, Signature Generation and Signature Verification. **Vendor Note:** Key sizes up to modulus length 8192-bit are supported for key generation, Signature Generation and Signature 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.	Used to support the following services: Self-test, Signature Generation (Public Key Cryptography) and Key Pair Generation. This implementation is used when **Partition Policy (16) Operate without RSA blinding** is disabled. This policy is enabled by default for all modules.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
RSA #2631.	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Method:** Signature Verification. **Signature Type:** PKCS #1-v1.5. **Hash options:** SHA1, SHA2-384.	**Modulus length:** 4096-bit.	Used to support the following services when using bootloader version 1.1.1: Request authentication and execution of main firmware, Self-test.
RSA #2632.	**Algorithm:** RSA. **Standard:** [FIPS 186-4].	**Method:** Signature Verification. **Signature Type:** PKCS #1-v1.5. **Hash options:** SHA1, SHA2-384.	**Modulus length:** 4096-bit.	Used to support the following services when using bootloaders 1.1.2, 1.1.4 and 1.1.5: Request authentication and execution of main firmware, Self-test.
DSA #1452.	**Algorithm:** DSA. **Standard:** [FIPS 186-4].	**Methods:** Parameter Generation, Key Generation, Signature Generation, Signature Verification. Hash options: Parameter Generation: SHA2-224, SHA2-256. Signature Generation: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification: SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Vendor affirmed using [FIPS 140-2 IG], A.11, 'The Use and the Testing Requirements for the Family of Functions defined in FIPS 202', when using SHA3.	**Modulus length:** 2048 and 3072 – Parameter Generation, Key Generation, Signature Generation. 1024, 2048 and 3072 – Signature Verification.	Used to support the following services: Key Pair Generation, Signature Generation (Public Key Cryptography) , Self-test and Signature Verification (Public Key Cryptography).

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
ECDSA #1526	**Algorithm:** ECDSA. **Standard:** [FIPS 186-4].	**Methods:** Key Generation, Signature Generation, Signature Verification. **Hash options:** Signature Generation: SHA2-224, SHA2-256, SHA2-384, SHA2-512. Signature Verification: SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521. Non-NIST (as per [FIPS 140-2 IG] IG A.2): see Table 10 below. **Vendor Note:** Signature Verification using curves B-163, K-163 and P-192 is listed as tested on CAVP cert ECDSA #1526 but where these curves are not available for use as approved.	Used to support the following services: Open Cloning Session / Clone Partition Object (CPV4)[22], Download Provider Domain Object[23], Open XTC Session[24], Self-test, Signature Generation (Public Key Cryptography) and Signature Verification (Public Key Cryptography).
#A1171	**Algorithm:** ECDSA. **Standard:** [FIPS 186-4].	**Methods:** Signature Generation, Signature Verification. **Hash options:** Signature Generation: SHA3-224, SHA3-256, SHA3-384, SHA3-512. Signature Verification: SHA3-224, SHA3-256, SHA3-384, SHA3-512.	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521. Non-NIST (as per [FIPS 140-2 IG] IG A.2): see Table 10 below.	Used to support the following services: Self-test, Signature Generation (Public Key Cryptography) and Signature Verification (Public Key Cryptography).

[22] service Open Cloning Session / Clone Partition Object (CPV4) exclusively uses curve P-521.
[23] service Download Provider Domain Object exclusively uses curve P-521.
[24] serviceOpen XTC Session exclusively uses curve P-521.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Key Agreement Scheme				
#A1171	**Algorithm:** KAS **Standard:** [SP800-56Ar3], [SP800-56Cr2].	**Methods:** dhHybrid1, dhEphem, dhHybridOneFlow and dhOneFlow with either: • 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; or • X9.42 KDF from [SP800-135r1] using SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	**Modulus length:** 2048, 3072 and 4096. **Caveat:** key establishment methodology provides between 112 and 201-bits of encryption strength. **Vendor Note:** Key sizes up to modulus length 8192-bit are supported for key derivation 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.	Used to support the following services: Self-test, Key Derivation.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A1171	**Algorithm:** KAS **Standard:** [SP800-56Ar3], [SP800-56Cr2], [FIPS 140-2 IG], [SP800-135r1].	**Methods:** ephemeralUnified and onePassDH with either: • OneStep KDF from [SP800-56Cr2][25] with Auxiliary function: SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. • OneStep KDF without a Counter[26] from FIPS 140-2 IG, D.14 with Auxiliary function: SHA2-256. • X9.63 KDF from [SP800-135r1] using SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. Vendor affirmed using [FIPS 140-2 IG], A.11, 'The Use and the Testing Requirements for the Family of Functions defined in FIPS 202', when using SHA3 with X9.63 KDF from [SP800-135r1].	**Curves:** B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521. **Caveat:** key establishment methodology provides between 112 and 256 bits of encryption strength.	Used to support the following services: Open Cloning Session / Clone Partition Object (CPV4)[27], Self-test, Key Derivation and Open XTC Session[28].

[25] available for use with the C_DeriveKey command used over the XTC channel for ICD.
[26] used with the XTC secure messaging channel exclusively.
[27] service Open Cloning Session / Clone Partition Object (CPV4) exclusively uses curve P-521 and One Step KDF from [SP800-56Cr2] with auxiliary function HMAC-SHA2-256.
[28] service Open XTC Session exclusively uses curve P-521 with One-Step KDF with No Counter from [FIPS 140-2 IG], D.14 with auxiliary function SHA2-512.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Key Transport				
AES #5652	**Algorithm:** KTS (Certs. AES #5652). **Standards:** [FIPS 197] and [SP800-38F].	**Modes:** GCM, KW and KWP.	**Key size:** 128, 192, and 256-bits. **Caveat:** key establishment methodology provides between 128 and 256 bit of encryption strength.	Used to support the following services: Open Cloning Session / Clone Partition Object (CPV4), Process Incoming/Outgoing XTC Traffic, Partition Export, Partition Import, Receive/Process JWE/JOSE Message for Device, Receive/Process JWE/JOSE Message for Domain, Self-test, Session Object Export, Session Object Import, Transfer Provider Domain Object, Unwrap Symmetric/Asymmetric Key, Wrap Asymmetric Key and Wrap Symmetric Key.
#A1171.	**Algorithm:** KTS-RSA. **Standards:** [SP800-56Br2] and [SP800-56Cr2].	**Method:** KTS-OAEP-basic. **Key generation method:** rsakpg1-crt and rsakpg2-crt. **Hash:** SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. **Mask Generation Function:** SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	**Modulus length:** 2048, 3072, 4096, 6144, and 8192. **Caveat:** key establishment methodology provides between 112 and 201 bits of encryption strength.	Used to support the following services: Encrypt/Decrypt (Asymmetric Algorithm), Receive/Process JWE/JOSE Message for Device, Receive/Process JWE/JOSE Message for Domain, Transfer Provider Domain Object, Download Provider Domain Object, Self-test, Unwrap Symmetric/Asymmetric Key and Wrap Symmetric Key. This implementation is used when **Partition Policy (16) Operate without RSA blinding** is enabled. This policy is enabled by default for all modules.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A1170.	**Algorithm:** KTS-RSA. **Standards:** [SP800-56Br2] and [SP800-56Cr2].	**Method:** KTS-OAEP-basic. **Key generation method:** rsakpg1-crt and rsakpg2-crt. **Hash:** SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. **Mask Generation Function:** SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	**Modulus length:** 2048, 3072, 4096, 6144, and 8192. **Caveat:** key establishment methodology provides between 112 and 201 bits of encryption strength.	Used to support the following services: Encrypt/Decrypt (Asymmetric Algorithm) and Self-test, Unwrap Symmetric/Asymmetric Key. This implementation is used when **Partition Policy (16) Operate without RSA blinding** is disabled. This policy is enabled by default for all modules.
Key Derivation Function				
#A1171.	**Algorithm:** KDA. **Standards:** [SP800-56Cr2], [FIPS 140-2 IG] D.14.	**Method:** One-Step Key Derivation **Hash:** SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512.	**Shared secret length:** 224-8192, increment 1 byte. **Derived Key length:** 128 – 4096-bits, increment 1 byte.	Used to support the following service: Clone Partition Object (CPV1), Open Cloning Session / Clone Partition Object (CPV4), Login and Key Derivation, Self-test.
#A1171.	**Algorithm:** KDA. **Standards:** [FIPS 140-2 IG] D.14.	**Method:** One-Step Key Derivation with No Counter **Hash:** SHA2-256.	**Shared secret length:** 384-bits. **Derived Key length:** 256-bits.	Used to support the following services: Open XTC Session, Self-test.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
KDF #234	**Algorithm:** Key-Based Key Derivation Function (KBKDF). **Standards:** [SP800-108].	**Mode:** Counter. **MAC Mode:** CMAC-AES128, CMAC-AES192, CMAC-AES256, CMAC Triple-DES, 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. Vendor affirmed using [FIPS 140-2 IG], A.11, 'The Use and the Testing Requirements for the Family of Functions defined in FIPS 202', when using SHA3.	**Supported Lengths:** 1024, 1032, 2048, and 2056. **Fixed Data Order:** Before Fixed Data. **Counter Length:** 32.	Used to support the following services: Open Cloning Session / Clone Partition Object (CPV4), Partition Export, Partition Import, Self-test, Session Object Export, Session Object Import and Key Derivation.
#A1171	**Algorithm:** KAS-ECC-SSC. **Standards:** [SP800-56Ar3].	**Methods:** ephemeralUnified, onePassDH.	**Curves:** P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-163, B-233, B-283, B-409, B-571.	Used to support the following services: Self-test, Open Cloning Session / Clone Partition Object (CPV4), Open XTC Session and Key Derivation.
#A1171	**Algorithm:** KAS-FFC-SSC. **Standards:** [SP800-56Ar3].	**Methods:** dhHybrid1, dhEphem, dhHybridOneFlow, dhOneFlow.	**Modulus length:** 2048, 3072 and 4096-bits. **Vendor Note:** Key sizes up to modulus length 8192-bit are supported for key derivation 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.	Used to support the following services: Self-test, Key Derivation.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A1171.	**Algorithm:** X9.42 Key Derivation Function (CVL). **Standards:** [SP800-133], [SP800-135r1] and [ANSI X9.42].	**Methods:** 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. **Derived Key Length:** 384-bits.	Used to support the following services: Key Derivation, Self-test.
#A1171.	**Algorithm:** X9.63 Key Derivation Function (CVL). **Standards:** [SP800-133], [SP800-135r1] and [ANSI X9.63].	**Methods:** SHA2-224, SHA2-256, SHA2-384, SHA2-512.	**Field Size:** 224, 256, 384, 521. **Shared Secret Length**: 128-4096 Increment 8-bits.	Used to support the following services: Encrypt/Decrypt (Asymmetric Algorithm), Self-test.
Random Number Generation				
DRBG #2283.	**Algorithm:** CTR_DRBG. **Standard:** [SP800-90Ar1].	**Mode:** AES-256.	**Security strength:** 256-bits.	Used to support the following services: Clone Partition Object (CPV1), Open Cloning Session / Clone Partition Object (CPV4), Create Domain, Create Partition, Download Provider Domain Object, Encrypt/Decrypt (Asymmetric Algorithm), Encrypt/Decrypt (Symmetric Algorithm), Retrieve DRBG output for export, Initialize Partition, Key Generation, Key Pair Generation, Self-test, Signature Generation (Public Key Cryptography), Transfer Provider Domain Object and Wrap Asymmetric Key.
N/A	**Algorithm:** ENT (P). **Standards:** [SP800-90B], [FIPS 180-4].	**Methods:** Live noise source with SHA2-512 vetted conditioning function.	**Security Strength:** Full Entropy.	Used to support the following service: Internal service used exclusively as a background service to periodically seed or reseed the platform and partition DRBG instances.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strength(s)	Use / Function
Key Generation				
Vendor Affirmed	**Algorithm:** CKG. **Standard:** [SP800-133].	**Method:** symmetric keys and seed for asymmetric key generation are created based on the direct output of the module DRBG (DRBG #2283).	**Security strength:** 256-bits.	Used to support the following services: Clone Partition Object (CPV1), Open Cloning Session / Clone Partition Object (CPV4), Create Domain, Create Partition, Download Provider Domain Object, Initialize Partition, Key Generation, Key Pair Generation and Transfer Provider Domain Object.

Table 9: Allowed Security Function for the Firmware Implementation

Algorithm	Description / Caveat	Use / Function
Key Transport		
AES #5652	AES (key unwrapping; key establishment methodology provides between 128 and 256 bits of encryption strength) (based on Cert. AES #5652 and using allowances in [FIPS 140-2 IG] D.9). **Vendor Note:** This allowed method supports the use of CBC, CTR and ECB modes with AES for key decryption exclusively for key import.	Used to support the following services: Unwrap Symmetric/Asymmetric Key and Clone Partition Object (CPV1).
Key Agreement		
#A1171	KAS-ECC-SSC (key establishment methodology provides between 112 and 285-bits of encryption strength). **Curves:** Non-NIST (as per [FIPS 140-2 IG] IG A.2): see Table 10 below.	Used to support the following services: Key Derivation.

NOTE ECB mode with AES is exclusively supported for decrypting symmetric keys and is not permitted for decrypting other importable key types.

Table 10: Supported non-NIST elliptic curve as per [FIPS 140-2 IG] IG A.2

Curve Name	Curve Field Type	Definition	Security Strength	Permitted Operations		
				Sign	Verify	Derive
sect571r2	Binary field - GF(2^m)	[SEC 2].	285-bits	x	x	x
sect571k2	Binary field - GF(2^m)	[SEC 2].	285-bits	x	x	x
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
X9.62 c2pnb431r1	Binary field - GF(2^m)	[ANSI X9.62].	215-bits	x	x	x
sect409r1	Binary field - GF(2^m)	[SEC 2].	204-bits	x	x	x
sect409k1	Binary field - GF(2^m)	[SEC 2].	204-bits	x	x	x
Brainpool P384r1	Prime field – GF(p)	[RFC 5639].	192-bits	x	x	x
Brainpool P384t1	Prime field – GF(p)	[RFC 5639].	192-bits	x	x	x
X9.62 c2pnb368w1	Binary field - GF(2^m)	[ANSI X9.62].	184-bits	x	x	x
X9.62 c2pnb359v1	Binary field - GF(2^m)	[ANSI X9.62].	179-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
X9.62 c2pnb304w1	Binary field - GF(2^m)	[ANSI X9.62].	152-bits	x	x	x
sect283r1	Binary field - GF(2^m)	[SEC 2].	141-bits	x	x	x
sect283k1	Binary field - GF(2^m)	[SEC 2].	141-bits	x	x	x
X9.62 c2pnb272w1	Binary field - GF(2^m)	[ANSI X9.62].	136-bits	x	x	x
sm2p256v1	Prime field – GF(p)	[ISO/IEC 14888-3:2018]	128-bits	x	x	x
secp256k1	Prime field – GF(p)	[SEC 2].	128-bits	x	x	x
Brainpool P256r1	Prime field – GF(p)	[RFC 5639].	128-bits	x	x	x
Brainpool P256r1	Prime field – GF(p)	[RFC 5639].	128-bits	x	x	x
Curve25519	Prime field – GF(p)	[RFC 7748]	128-bits	x	x	x
X9.62 prime239v3	Prime field – GF(p)	[ANSI X9.62].	119-bits	x	x	x
X9.62 prime239v2	Prime field – GF(p)	[ANSI X9.62].	119-bits	x	x	x
X9.62 prime239v1	Prime field – GF(p)	[ANSI X9.62].	119-bits	x	x	x
X9.62 c2pnb239v1	Binary field - GF(2^m)	[ANSI X9.62].	119-bits	x	x	x
X9.62 c2pnb239v2	Binary field - GF(2^m)	[ANSI X9.62].	119-bits	x	x	x
X9.62 c2pnb239v3	Binary field - GF(2^m)	[ANSI X9.62].	119-bits	x	x	x

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

2.9.2 Non-Approved Algorithm Implementations

Non-FIPS Approved security functions are not available for use when the module has been configured to operate in FIPS-approved mode. See section 3.1, 'Identifying the Module Version'.

> **Symmetric Encryption/Decryption**

- DES;
- RC2;
- RC4;
- RC5;
- CAST3;
- CAST5;
- SEED;
- ARIA; and
- SM4.

> **Hashing**

- MD2;
- MD5;
- HAS-160; and
- SM3.

> **Message Authentication Code**

- AES MAC;
- DES-MAC;
- RC2-MAC;
- RC5-MAC;
- CAST3-MAC;
- CAST5-MAC;
- SEED-MAC;
- ARIA-MAC;
- SSL3-MD5-MAC;
- SSL3-SHA1-MAC; and
- HMAC (non-compliant for any configuration providing less than 112 bits of encryption strength).

> **Asymmetric**

- KCDSA;
- RSA X-509;
- RSA (non-compliant with less than 112 bits of encryption strength);

- RSA (non-compliant for key transport using non-[SP800-56Br2] schemes);
- DSA (non-compliant with less than 112 bits of encryption strength);
- ECDSA (non-compliant with less than 112 bits of encryption strength);
- Deterministic ECDSA; and
- EdDSA.

> **Key Generation**

- DES;
- RC2;
- RC4;
- RC5;
- CAST3;
- CAST5;
- SEED;
- ARIA;
- GENERIC-SECRET;
- X9-42 Domain Parameter Generation; and
- BIP32.

> **Key Agreement**

- ECC (non-compliant with less than 112 bits of encryption strength); and
- Diffie-Hellman (key agreement; key establishment methodology; non-compliant with less than 112 bits of encryption strength).

> **Key Transport**

- RSA (key wrapping; key establishment methodology; non-compliant with less than 112 bits of encryption strength).

2.10 Critical Security Parameters

The following table lists Critical Security Parameters (CSP) used to perform approved security function supported by the cryptographic module:

Table 11: Summary of CSPs

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Password (Authentication Data) – PSO, PCO, PCU	7 - 255 character data string	N/A – supplied by user during initialization of target role.	Input from host using Luna ICD communication path over XTC.	User provided password input by the operator as authentication data.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
PED Authentication Data	48-byte random value	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Input / output via direct connection to the Thales Luna PED.	Random value that is generated by the module when a role is created and is written out to the iKey via the Trusted Path. This CSP is used as part of authenticating the PED authenticated PDA role.
Password (Authentication Data) – CVM	16-byte	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Copy output on creation encapsulated in a JWE encrypted package under a client generated CEK.	A 16-byte random value generated by the module that is base64 encoded and returned to the PDA on creation of a Cryptovisor Manager (CVM).
Key Cloning Domain Vector (KCV)	7 - 255 character data string or 48-byte value	N/A.	Option 1: Password entered by user over XTC; or Option 2: 32-byte secret read from the Thales Luna PED and entered over XTC.	Used as an input to the cloning protocol (CPV1 and CPV4). It can either be submitted as a character string by the user or read from a red iKey where this is connected to a Thales Luna PED connected to the host machine for the Thales Luna client. This key is exclusively imported to the module where it does not generate this CSP.
Token Unwrapping Key 3 (TUK3)	RSA 2048 bit private key	[FIPS 186-4] RSA Key Generation [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Not input or output.	A 2048 bit RSA private key used with CPV1.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Token Wrapping Key 3 (TWC3)	RSA-2048 bit public key certificate	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Certificate output in plaintext	The X.509 public key certificate corresponding to the TUK3. It is created and signed by the HOK the first it is required. Used as part of CPV1.
Cloning Key Encryption Vector – source (KEV_s)	384 bit nonce.	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Exchanged during CPV1 protocol encrypted under TWC3 using PKCS#1 v1.5 Encryption. This is separate then encrypted using AES-GCM and the XTC Partition Tunnel Key.	384 bit nonce used with CPV1 and generated on the source HSM.
Cloning Key Encryption Vector – target (KEV_t)	384 bit nonce.	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Exchanged during CPV1 protocol encrypted under TWC3 using PKCS#1 v1.5 Encryption. This is separately then encrypted using AES-GCM and the XTC Partition Tunnel Key.	384 bit nonce used with CPV1 and generated on the target HSM.
Cloning Transfer Key	AES-256	Derived from the KEV_t, ,KEV_s and the KCV using OneStep KDF from [SP800-56Cr2] with SHA2-512 as the PRF.	Not input or output.	256-bit AES key derived during CPV1 and used to transfer key objects between source and target partitions using the cloning protocol.
User Storage Key (USK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Not input or output.	This key is used to encrypt all sensitive attributes of all private objects owned by a user and contained in a User Partition.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Security Officer Master Key (SMK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Not input or output.	This key is used to encrypt all sensitive attributes of all private objects owned by the Provider Domain Administrator (PDA) and stored in the Admin Partition.
Partition Storage Key (PSK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Not input or output.	This key is unique per-partition and used to encrypt all CSP that are shared by all roles of a given partition.
Global Storage Key (GSK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Not input or output.	32-byte AES key that is the same for all users on a specific module. It is used to encrypt permanent parameters within the non-volatile memory area reserved for use by the module.
Root Certificate (ROOT)	RSA-4096 public key certificate	N/A: Loaded at manufacturing	Certificate embedded in the bootloader, which is loaded at manufacture. Certificate output in plaintext.	The X.509 public key certificate corresponding to the Root Key. It is self-signed. Used in verifying Manufacturing Integrity Certificate (MIC) and firmware updates. The certificate can be requested from the module by the end-user where it is exported in plaintext.
Manufacturer's Integrity Certificate (MIC)	RSA-4096 public key certificate	N/A: Loaded at manufacturing	Certificate loaded in plaintext at manufacture. Certificate output in plaintext.	The X.509 public key certificate corresponding to the Manufacturing Integrity Key (MIK). It is signed by the Root Key. Used in verifying Hardware Origin Certificates (HOCs). The certificate can be requested from the module by the end-user where it is exported in plaintext.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
ECC Manufacturer's Integrity Certificate (ECC MIC)	EC-secp384r1 public key certificate	N/A: Loaded at manufacturing	Certificate loaded in plaintext at manufacture. Certificate output in plaintext.	The X.509 public key certificate corresponding to the ECC Manufacturing Integrity Key (ECC MIK). Certificate is self-signed. Used in verifying ECC HOC. The certificate can be requested from the module by the end-user where it is exported in plaintext.
Hardware Origin Key (HOK)	RSA 4096 bit private key	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Not input or output.	A 4096-bit RSA private key used to sign device messaging, domain origin, and partition-messaging key. Used for signing join-request message (RSASSA-PKCS1-v1_5 using SHA2-384) It is generated at the time the device is manufactured.
Hardware Origin Certificate (HOC)	RSA-4096 public key certificate	N/A: Loaded at manufacturing	Certificate loaded in plaintext at manufacture. Certificate output in plaintext.	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.
ECC Hardware Origin Key (ECC HOK)	EC-secp384r1 private key	[FIPS 186-4] ECC Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Not input or output.	A 384-bit EC private key – the key is used to generate proof of possession in relation to ECC-HOC.
ECC Hardware Origin Certificate (ECC HOC)	EC-secp384r1 public key certificate	N/A: Certificate loaded at Manufacture – public key generated using same process as ECC HOK.	Certificate loaded in plaintext at manufacture. Certificate Output in Plaintext.	The X.509 public key certificate corresponding to the HOK. It is signed by the ECC Manufacturer's Integrity Key (ECC MIK)

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Token Module Variable Key (TVK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Not input or output.	Stored in HSE-BBRAM and used to cache PED pin when the auto-activation policy is enabled. This policy is not supported in Cryptovisor and although the key is present, it is redundant.
Password Encryption Key (PEK)	RSA 4096 bit private key	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Not input or output.	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 and is not persistent over a power-cycle.
Password Encryption Certificate (PEC)	RSA-4096 public key certificate	[FIPS 186-4] RSA Key Generation [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Certificate output in plaintext	The X.509 public key certificate corresponding to the PEK. It is created and signed by the HOK the first it is required.
Firmware Signing Cert (FSC)	RSA-4096 bit public key certificate.	N/A: Delivered with Firmware Update Package.	Certificate input in plaintext as part of Firmware update File (FUF).	The X.509 public key certificate corresponding to the Firmware Signing Key (FSK). It is signed Thales Root signing key. Used to verify Firmware images on initial load and subsequently on power-on.
Licensing Signing Cert (LSC)	RSA-4096 bit public key certificate.	N/A: Delivered with Configuration Update Package.	Certificate input in plaintext as part of Configuration update File (CUF).	The X.509 public key certificate corresponding to the License Signing Key (LSK). It is signed Thales Root signing key. Used to verify CUF on load.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Manufacturer Authentication Certificate (MAC)	RSA-2048 bit public key certificate	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Certificate output in plaintext.	The X.509 public key certificate corresponding to the Manufacturer Authentication Key (MAK). It is self-signed using the MAK. Used in verifying DAK.
ECC Manufacturer Authentication Certificate (ECC MAC)	EC-secp384r1 public key certificate	N/A: Loaded at manufacturing.	Certificate output in plaintext.	The X.509 public key certificate corresponding to the Manufacturer Authentication Key (ECC MAK). It is self-signed using the ECC MAK. Used in verifying ECC DAK.
Device Authentication Key (DAK)	RSA-2048 private key	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Not input or output.	2048-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.
Device Authentication Certification (DAC)	RSA-2048 public key certificate	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Certificate output in plaintext.	The X.509 public key certificate corresponding to the DAK. It is signed by the HOK (private key corresponding to HOC). Used for a specific PKI implementation requiring assurance that a key or a specific action originated within the hardware crypto module.
ECC Device Authentication Key (ECC DAK)	EC secp384r1 private key	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Not input or output.	The X.509 public key certificate corresponding to the ECC DAK. It is signed by the ECC HOK.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
ECC Device Authentication Certificate (ECC DAC)	EC secp384r1 public key certificate	Public key generated using same process as ECC DAK.	Certificate output in plaintext.	ECC P-384 private key used for a specific PKI implementation requiring assurance a specific action originated within the hardware crypto module. It is signed by the ECC MAK (private key corresponding to MAC).
Key Encryption Key (KEK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Not Input or Output.	The KEK encrypts all sensitive values and is zeroized in response to a decommission signal.
DRBG Key	AES-256	ENT (P) approved noise source.	Not Input or Output.	32 bytes AES key stored in the RAM. Used in an implementation of CTR_DRBG from [SP800-90Ar1].
DRBG Seed	384 bits	ENT (P) approved noise source.	Not input or output.	384-but random seed/re-seed data drawn from the Hardware RBG and used to seed the platform DRBG.
DRBG V	128 bits	ENT (P) approved noise source.	Not input or output.	Part of the secret state of the approved DRBG. The value is generated using the methods described in [SP800-90Ar1] for CTR_DRBG.
Domain Origin Key (DOK)	RSA-4096	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Input/ Output using AES-GCM as part of PDO. Encrypted under Domain Transport Key.	Key signing for domain and partition key hierarchy for origin and messaging certificates.
Domain Origin Cert (DOC)	RSA-4096	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Certificate output in plaintext.	Used to Authenticate the Domain Messaging Key alongside XTC – Partition Messaging Key

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Domain Messaging Key (DoMK)	RSA-4096	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Input/ Output using AES-GCM as part of PDO. Encrypted under Domain Transport Key.	Used for secure communication between client and HSM domain where the key is used with KTS-OAEP-basic [SP800-56Br2].
Domain Messaging Cert (DoMC)	RSA-4096	Generated by the HSM based on the Domain Messaging Key.	AES-GCM encrypted under JOSE response key.	Used for secure communication between client and HSM domain where the key is used with KTS-OAEP-basic [SP800-56Br2].
Device Messaging Key (DeMK)	RSA-4096	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Not input or output.	Used for JOSE/JWE messaging API to communicate between client and HSM where the key is used with KTS-OAEP-basic [SP800-56Br2].
Device Messaging Cert (DeMC)	RSA-4096	Generated by the HSM based on the Device Messaging Key.	Certificate output in plaintext	Verified by the client. Used for JOSE/JWE messaging API to communicate between client and HSM where the key is used with KTS-OAEP-basic [SP800-56Br2].
Domain Encryption Master Key (DEK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Input / Output using AES-GCM under Join-Protocol -PDO Transfer Key when the PDO it transferred between HSM using the Join Protocol.	Used as key derivation key for encryption of partitions, sessions, and containers for export

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Join-Protocol (Target) Content Encryption Key (JP(T)-CEK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Import/Export during Join protocol encrypted using KTS-OAEP-Basis from [SP800-56Br2] with SHA2-384 MGF under the Domain Messaging Cert.	Used to encrypt and authenticate join request messages (AES-GCM).
Join-Protocol - PDO Transfer Key (JP-PTK)	AES-256	Generated by Target Device in join-protocol using [SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Exported from target device for Join protocol encrypted using KTS-OAEP-Basis from [SP800-56Br2] with SHA2-384 MGF under the Domain Messaging Cert. Imported by Root Device by decrypting with Domain Messaging Key.	Used as the single-use symmetric key used to transfer the PDO between cryptographic modules where both devices appear on the ADL.
Join-Protocol (Root) content encryption key (JP(R)-CEK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher when on Root device.	Imported to Session device during Join protocol encrypted using KTS-OAEP-Basis from [SP800-56Br2] with SHA2-384 MGF under the Domain Messaging Cert.	Used to encrypt and authenticate join response messages (AES-GCM).
Partition Export Key (ParEK)	AES-256	Derived from Domain Encryption Key using Counter KDF from [SP800-108] with AES-CMAC as the PRF.	Input/output using AES-GCM as part of PDO. Encrypted under Domain Transport Key.	Used to export the partition header.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Partition Fragment Key (PFK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Input/output using AES-GCM under the Partition Export Key.	This encrypts a partition's objects and sessions.
XTC Partition Messaging Key (XTC-PMK)	EC secp384r1 private key	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	Input/output using AES-GCM under the Partition Export Key.	The HSMs static private key used in ECDH negotiation of the XTC tunnel transport key.
XTC Partition Messaging Certificate (XTC-PMC)	EC secp384r1 public key certificate	[FIPS 186-4] RSA Key Generation using [SP800-90Ar1] CTR_DRBG with AES256 cipher for raw entropy.	AES-GCM encrypted under JOSE response key.	The HSMs static public key used in ECDH negotiation of the XTC tunnel transport key.
XTC ephemeral client key (XTC-epCK)	EC secp384r1 public key	N/A – Generated by Thales Luna Client	Input using AES-GCM encrypted under JOSE Content Encryption Key.	The clients ephemeral key pair used in ECDH negotiation of the XTC tunnel transport key.
XTC tunnel key agreement key	256-bit generic secret	ECDH (1e, 1s) from [SP800-56Ar3].	No input/output – single use ephemeral key.	Intermediary CSP during the XTC key exchange. This key is only used to feed the KDF used to derive the XTC tunnel transport key.
XTC tunnel transport key (XTC-TTK)	AES-256	OneStep KDF_NoCounter from [FIPS 140-2 IG], D.14 with SHA2-512 as the PRF.	No input/output – single use ephemeral key.	Used to transport the XTC partition tunnel key and partition token to the user.
XTC tunnel key derivation key (XTC-TDK)	256-bit generic secret	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Input/output using AES-GCM encrypted under the Partition Export Key.	Used in key derivation of XTC partition tunnel key.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
XTC Partition Tunnel Key (XTC-PTuK)	AES-256	OneStep KDF_NoCounter from [FIPS 140-2 IG], D.14 with SHA2-512 as the PRF.	Transmitted to client using XTC tunnel transport key.	Protects ICD commands between client and HSM.
XTC Partition Token Key (XTC-PToK)	AES-256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Input/output using AES-GCM encrypted under the Partition Export Key.	Used to sign XTC token.
XTC Partition Token (XTC-PT)	Token (blob)	[SP800-90Ar1] CTR_DRBG with AES-256 cipher for token nonce and system time for timestamp	Transported in plain text in the XTC header.	Used as public context to device XTC partition tunnel key.
XTC secret AppID (XTC-SA)	256-bit generic secret	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Transmitted encrypted using AES-GCM as part of XTC tunnel under XTC Partition Tunnel Key.	Proves access to a session group.
JOSE response key (JOSE-RK)	AES-256	N/A – Generated by Thales Luna Client.	Input using KTS-OAEP-Basis from [SP800-56Br2] under either Device, Domain or Partition Messaging Certificate.	Encrypts responses to JOSE/JWE API as used for the Domain Administration API.
JOSE Content Encryption Key (JOSE-CEK)	AES-256	N/A – Generated by Thales Luna Client	Input using KTS-OAEP-Basis from [SP800-56Br2] under either Device, Domain or Partition Messaging Certificate.	Encrypts responses to JOSE/JWE API as used for the Domain Administration API.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Asymmetric Key Pairs (general partition or session keys)	RSA, DSA, ECC, DH	Option 1: N/A (user imported); or Option 2: [FIPS 186-4], Appendix B.4.1. – for module generated ECC key-pair using [SP800-90Ar1] CTR_DRBG with AES-256 cipher for seed material; or Option 3: [FIPS 186-4], Appendix B.3.6. – for RSA, DH and DSA keys using the [SP800-90Ar1] CTR_DRBG with AES-256 cipher for seed material.	Input encrypted using Symmetric Keys (general partition or session keys) where loaded by user / Output encrypted using Symmetric Keys (general partition or session keys).	General use asymmetric key pairs that can be exported/imported from/to the module or generated by the module.
Symmetric Keys (general partition or session keys)	AES (including AES-XTS), MAC, KDF	Option 1: N/A (user imported); or Option 2: [SP800-90Ar1] CTR_DRBG with AES-256 cipher. (module generated)	Input encrypted using Asymmetric or Symmetric Key Pairs (general partition or session keys) where loaded by user / Output encrypted using either Asymmetric or Symmetric Key Pairs (general partition or session keys).	General use symmetric keys that can be exported/imported from/to the module or generated by the module.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
CPV4 Cookie Key Derivation Key	256-bit generic secret	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Input /output using AES-GCM under Join-Protocol -PDO Transfer Key when the PDO it transferred between HSM using the Join Protocol.	Used to derive the cookie key CPV4.
CPV4 Cookie Key	AES-GCM 256	Derived from PFK, Session ID and CPV4 Cookie Key Derivation Key using counter KDF from [SP800-108] with HMAC-SHA2-256 as the PRF.	No input/output – single use ephemeral key.	The cookie key persists the CPV4 state. The state includes the CPV4 session key.
CPV4 Key Agreement Private Key	EC secp521r1 private key	FIPS PUB 186-4, Appendix B.4.1.	Output stored in the session cookie encrypted using the CPV4 Cookie Key. The key is encrypted using AES-256 in GCM mode with a 128-bit random IV and 128-bit Tag.	Used to establish the ECDH key agreement between HSMs.
CPV4 Key Agreement Public Key	EC secp521r1 public key	N/A – imported from peer HSM.	Imported from a peer HSM in plaintext as part of the signed NegotiateResponse message.	Used to establish the ECDH key agreement between HSMs.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
CPV4 Key Agreement Shared Secret	256-bit generic secret	Derived using ECDH OneStep Key Derivation (section 5.8.2.1 from [SP800-56Ar3]) with C(2e, 0s, ECC CDH) with P-521.	No Input/output. Single use ephemeral key.	Shared secret output from ECDH during CPV4 negotiation. Shared secret is used to derive the CPV4 Key Transport Key.
CPV4 Key Transport Key	AES-GCM 256	Derived using OneStep KDF from [SP800-56Cr2] and SHA2-512 as the PRF. Inputs to the KDF include the CPV4 Key Agreement Shared Secret alongside the Key Cloning Domain Vector (KCV).	No Input/output. Single use ephemeral key.	This wraps the session key used to encrypt the cloning protocol 4 session messages.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
CPV4 Session Key	AES-GCM 256	[SP800-90Ar1] CTR_DRBG with AES-256 cipher.	Option 1: Exported as part of the CPV4 protocol encrypted under the CPV4 Transport Key using AES-256 in GCM and with a 128-bit random IV and 128-bit TAG. Option 2: Imported/Exported stored in the session cookie encrypted using the CPV4 Cookie Key. The key is encrypted using AES-256 in GCM mode with a 128-bit random IV and 128-bit Tag.	This is the session key used to encrypt messages in CPV4. The key is directly used to encrypt keys in transit between source and destination partitions.
EC Attestation Key	EC secp521r1 private key	[FIPS 186-4], Appendix B.4.1. – for module generated ECC key-pair using the [SP800-90Ar1] CTR_DRBG with AES-256 cipher for seed material	Input /output using AES-GCM under Join-Protocol -PDO Transfer Key when the PDO it transferred between HSM using the Join Protocol. [SP800-90Ar1]	This signs CPV4 messages and the ephemeral keys. It is signed by the Domain Origin Key.

2.10.1 Key Generation

Symmetric cryptographic keys are generated by the direct unmodified output of the module's [SP800-90Ar1] DRBG. The DRBG output is also used as a seed for asymmetric key generation.

Keys that are generated outside the module and input during the manufacturing process include: Root Certificate, MIC and ECC MIC.

The HOC and ECC HOC are created outside the module during manufacture based on public keys generated by the module and exported as part of the manufacturing process. Once signed by corresponding externally managed keys, these are re-loaded onto the module for subsequent validation and storage.

User passwords for authentication of the PSO, PCO and PCU roles are generated by the operator.

User passwords for authentication of the CVM role is generated by base64 encoding by the direct unmodified output of the module's [SP800-90Ar1] DRBG.

The [SP800-90Ar1] DRBG (CTR_DRBG using AES256) is seeded using a 384-bit full entropy seed from the platform Non-Deterministic Random Number Generator as covered in more detail in section 2.10.2, 'Non-Deterministic Random Number Generation Specification'.

2.10.2 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 an approved conditioning function where in-turn the output of the conditioning function is used to seed the DRNG (DRBG #2283).

The Non-Deterministic RNG complies with [SP800-90B] and has been certified using [FIPS 140-2], 7.18 and using guidance set out in IG 7.19.

Table 12: Non-Deterministic Random Number Generation Specification

Entropy sources	Minimum number of bits of entropy	Details
Non-deterministic jitter in from FRO.	Full-entropy output	[SP800-90Ar1] compliant Non-Deterministic RNG using a hardware based noise internal to the module boundary. Digitized output from the noise source is fed through an approved conditioning function based on SHA2-512 (SHS #4533). Raw noise is generated based on non-deterministic jitter built up in free-running oscillators. The module achieves full entropy from the output of the conditioning function where for every 384-bits used to seed the DRBG this includes 384-bits of entropy. All outputs from the noise source are subjected to statistical testing ahead of being fed to the conditioning function. The output of the hardware noise source includes a total failure test to check for bit-patterns consistent with hardware failures.

2.10.3 Key Import and Export

For details of encryption and specific CSP used, refer to Input/Output column of Table 11, 'Summary of CSPs' in section 2.10, 'Critical Security Parameters'.

Depending on the configuration of the module, the following methods of key entry and output are available as a service (see section 2.5.2, 'Services'):

> **Key Wrap / Unwrap using Cloning over XTC**

Key cloning is a product feature that supports the secure transfer of partition objects between different partitions, which can reside on the same or different modules. Objects transferred using the cloning protocol may be keys, user data, or module data. Two versions of the cloning protocol are supported with a description of each provided below:

- **CPV1** - uses AES-256 in CBC mode with a single-use key to encrypt an object being transferred. The single-use AES key is obtained by combining the 48-byte key cloning domain vector (KCV) (randomly

generated by the module on creation of a partition) with random one-time data generated by source and target cryptographic modules and exchanged using RSA 2048-bit key transport. The 'One-Step Key Derivation' function from [SP800-56Cr2] is used with SHA2-512 as the auxiliary function to generate the transport key.

All ICD messages involved in the transfer using Key Cloning are also independently encrypted as part of the XTC transport tunnel using AES-GCM and the XTC Partition Tunnel Key. Use of AES-GCM to super-encrypt the object in transit provides compliance with [SP800-38F].

- **CPV4** – uses AES-256 in GCM mode with a session based transfer key. The session key is agreed between source and destination modules using ECDH One-Step Key Derivation (section 5.8.2.1 from [SP800-56Ar3]) with C(2e, 0s, ECC CDH) with P-521 and using One-Step KDF from [SP800-56Cr2] with HMAC-SHA2-256 as the PRF to generate the resulting CPV4 Session Key.

> **Key Wrap / Unwrap over XTC**

The key wrap operation encrypts either a symmetric key or an asymmetric private key value for output, using either an RSA public key and RSA-OAEP or a symmetric key and AES (KTS).

The unwrap operation takes as input an encrypted symmetric key or asymmetric private key and a handle to the key that was originally used to do the wrapping. It decrypts the key, stores it in the module as a key object and returns the handle to the imported key.

Note that for both wrap and unwrap operations, the user (or calling application acting on the user's behalf) never has access to the actual key values, only handles assigned to the key objects in the module.

All ICD messages involved in the transfer using Key Wrap / Unwrap are independently also encrypted as part of the XTC transport tunnel using AES-GCM and the XTC Partition Tunnel Key generated using ECDH (1e,1s) from [SP800-56Ar3] and One-Step KDF with No Counter from [FIPS 140-2 IG], D.14. Use of AES-GCM provides compliance with [SP800-38F].

Where AES is used to unwrap keys using allowances permitted under [FIPS 140-2 IG], D.9, CBC, CTR and ECB modes exclusively are supported for use.

> **PDO Transfer using Join Protocol**

The Join Protocol is a product feature implemented using a sequence of commands from the Domain Administration API. This protocol is used to transfer the PDO between cryptographic modules pre-registered to a Domain based on the Domain ADL. Transfer of the PDO is performed using AES-GCM for encryption using JP-PTK, which itself is transferred between cryptographic modules using RSA-OAEP and the public key from the Domain Messaging Cert.

> **Key Wrap/Unwrap over Domain Administration API**

All sensitive CSP transferred using the Domain Administration API are encrypted as part of the JOSE/JWE request and response messages.

Any CSP imported to the module using this path are encrypted during input using the JOSE-CEK with AES-256 in GCM mode.

Any CSP returned by the module are encrypted using the JOSE-RK with AES-256 with GCM.

The JOSE-RK is transferred with each JWE/JOSE request message encrypted using RSA-OAEP and the 4096-bit public key corresponding to either the HOC or the DMC depending on whether the domain or an individual cryptographic module is the target for a given command.

> **Partition Import/Export over Domain Administration API**

Complete partitions may be inserted or extracted from the module encrypted using AES-GCM and either ParEK or PFK). These keys are part of the PDO and partitions can only inserted into a cryptographic module that is part of the domain the original partition was registered.

ParEK and PFK are never available in a plaintext form outside the cryptographic module. Exported partitions can only be inserted onto an HSM containing a copy of the PDO used for the domain from which the partitions were exported.

2.11 Self-Tests

2.11.1 Power-On Self-Tests

The module performs Power-On Self-Tests (POST) upon power-up to confirm the firmware integrity as well as to check the random number generator and each of the implemented cryptographic algorithms. While the module is running POST all interfaces are disabled until the successful completion of the self-tests. If any POST fails an error message is output, the module halts, and data output is inhibited.

These self-tests can also be initiated as an operator service but do not require operator input to initiate at power on.

Table 13: Power On Self-Tests (Bootloader) – Module Integrity

Test	When Performed	Indicator
SHA (SHA1 and SHA2-384) KAT (digest tests).	Power-on	Error output and module halt
RSA (4096-bit modulus) KAT (sign and verify tests).	Power-on	Error output and module halt
Boot loader performs an RSA 4096-bit SHA2-384 signature verification of itself.	Power-on	Error output and module halt
Boot loader performs an RSA 4096-bit SHA2-384 signature verification of the firmware prior to firmware start	Power-on/Request[29]	Error output and module halt

Table 14: Power On Self-Tests (Firmware) – Cryptographic Implementations

Test	When Performed	Indicator
DRBG Self-Test (Instantiate Function Known Answer Test, Generate Function KAT, Reseed Function KAT, conditional tests)	Power-on/once every 24 hours	Error output and module halt
SHA KAT (SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE-128, SHAKE-256)	Power-on/Request	Error output and module halt
HMAC KAT (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)	Power-on/Request	Error output and module halt
RSA KAT (Signature Generation, Sig Verification, Encrypt and Decrypt)	Power-on/Request	Error output and module halt

[29] Request indicates triggering a POST via a command

Test	When Performed	Indicator
DSA KAT (Signature Generation, Sig Verification)	Power-on/Request	Error output and module halt
Diffie-Hellman KAT (X9.42 derive operation – no KDF)	Power-on/Request	Error output and module halt
AES KAT (ECB, CBC, OFB, CFB, CFB128, CFB8, KW, KWP, GCM, XTS modes covering 128, 192 and 256 bit keys, CMAC and GMAC)	Power-on/Request	Error output and module halt.
ECDH KAT (Derive operation – no KDF)	Power-on/Request	Error output and module halt
ECDSA KAT (Signature Generation, Sig Verification)	Power-on/Request	Error output and module halt
KBKDF KAT (SP 800-108 KBKDF using AES-CMAC, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 as PRF)	Power-on/Request	Error output and module halt
KDF KAT (SP 800-56Cr2 One-Step KDF using SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512. IG D.14 One-Step KDF using no counter with SHA2-256. X9.42/X9.63 KDF using SHA1.	Power-on/Request	Error output and module halt

NOTE Self-tests for SP800-56Ar3 compliant key derivation operations are covered in Table 14as 'Diffie-Hellman KAT (X9.42 derive operation – no KDF)' and 'ECDH KAT (Derive operation – no KDF)'.

For each test, the cryptographic module performs a complete derive operation based on fixed inputs and compares the result against a stored result generated by an independent implementation.

The module implements an independent KAT for both the ECC Diffie-Hellman option and then separate for the FFC Diffie-Hellman operation and where the operation performed includes as a sub-component the required KAT defined under [FIPS 140-2 IG], D.8, Key Agreement Methods.

2.11.2 Conditional Self-Tests

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

Table 15: Conditional Self-Tests

Test	When Performed	Where Performed	Indicator
HRNG conditional tests (Total failure test on the output from the hardware noise source, Repetition Count Test and Adaptive Proportion Test statistical tests.) [30]	Continuous	Firmware / Hardware	Error output and module halt
RSA – Pair-wise consistency test (asymmetric key pairs)	On generation	Firmware	Error output
DSA – Pair-wise consistency test (asymmetric key pairs)	On generation	Firmware	Error output
ECC – Pair-wise consistency test (asymmetric key pairs – covers keys used for both ECDSA and ECDH)	On generation	Firmware	Error output
Diffie-Hellman - FFC full public-key validation ([SP800-56Ar3]/[SP800-56Br2] compliant implementation)	Ahead of public key use for derive operation.	Firmware	Error output
ECDH – ECC full public-key validation ([SP800-56Ar3] compliant implementation)	Ahead of public key use for derive operation.	Firmware	Error output
Firmware load test (4096-bit RSA Signature Verification)	On firmware update load	Firmware	Error output – module will continue with existing firmware

2.12 Mitigation of Other Attacks

Timing attacks are mitigated directly by a module through the use of hardware accelerator chips for modular exponentiation operations. The use of constant timing hardware acceleration ensures that all RSA signature operations complete in the same time, therefore making the analysis of timing differences irrelevant. RSA blinding may also be selected as an option.

[30] CRNGT, as described in Section 4.9.2 of FIPS 140-2, is only performed for the NDRNG and is not performed for the DRBG as permitted by [FIPS 140-2 IG], 9.8 for modules implementing an approved DRBG from [SP800-90Ar1].

3 Guidance

3.1 Identifying the Module Version

Ahead of putting the module into its approved mode of operation, it is important to identify the hardware, firmware and bootloader versions of the target module and to check these correspond to those listed in section 1.2, ‘Scope’. The following sections provide guidance on checking each element.

Any module returning hardware, firmware and bootloader versions not listed in this security policy is out of the scope of this validation and requires a separate FIPS 140-2 validation.

3.1.1 Checking the Bootloader Version

The bootloader version can be checked by viewing the output from `dmesg`, which can be run on the Linux based host platform following boot of the cryptographic module.

To find the bootloader version run the command:

```
dmesg | grep “Boot Loader 2” | tail –n 1
```

The bootloader version will be listed on a line similar to below:

```
26.145018] k7pf1: [hsm] Boot Loader 2 Revision K7 1.1.4.
```

3.1.2 Checking the Firmware Version

Two paths are supported to checking the firmware version depending on whether the user is local to the cryptographic module or connecting remotely over XTC.

For local users, the firmware version can be checked by running the `get-device-info` Domain Administration API, JOSE/JWE, command. The cryptographic module will return the firmware version, hardware serial number, device time and device state.

Example output returned by the command is shown below:

```
Firmware version: 2.0.2
Hardware serial number: 552649
Device time: Mon Jun 10 13:53:52 2021 (UTC) and 192845 microseconds
Device state: zeroized
```

For remote users connecting over XTC, the LunaCM, `slot list` command can be used where the 'CV firmware version' is listed in the output.

Example output from this command is shown below:

```
lunacm:>slot list
Slot Id -> 3
Label -> Cryptovisor Demo Partition
Serial Number -> 44732604723
Model -> Cryptovisor K7
```

```
Firmware Version -> 7.3.0
CV Firmware Version -> 2.0.2
Configuration -> Luna User Partition With SO (PW) Key Export Mode
Slot Description -> User Token Slot
Current Slot Id: 3
Command Result : No Error
```

The firmware version of importance is the 'CV Firmware Version', which embodies all firmware running on the cryptographic module. The 'Firmware Version' shown above as '7.3.0' is a separate version relating to a sub-set of the certified firmware only and should be ignored for the purposes of establishing the FIPS approved mode of operation.

3.1.3 Checking the Hardware Platform Identifier

The hardware version is stored in EEPROM on the cryptographic module during manufacture. The hardware identifier stored is read by the module and displayed during execution of the following commands `hsm showinfo` and `partition showinfo` LunaCM commands.

Output in response to the command will include a line listing the hardware part number. An example line showing one of the valid hardware platform identifiers for a module with hardware compliant with this security policy is shown below:

```
HSM Part Number -> 808-000048-002
```

3.2 Approved Mode of Operation

The cryptographic module is approved when running a FIPS 140-2 certified version of firmware as listed in section 1.2, 'Scope'.

To place the module in FIPS 140-2 Approved mode as defined by FIPS PUB 140-2, the PDA must set the "mode" flag to "fips" when executing the `'create-domain'` Domain Administrator API, JOSE/JWE, command at domain initialization.

Selecting this option will constrain any cryptographic module registered to the domain to only allow approved services.

Following creation of a compliant domain[31] – the status of the mode can be checked by calling the `'get-domain-info'` Domain Administrator API command that will list the mode as "fips".

As an alternative path to confirm the module is in FIPS mode over XTC, enter the `partition showinfo` command from LunaCM. As part of status information returned, the module will output the following statement confirming FIPS 140-2 approved mode of operation:

```
*** The HSM is in FIPS 140-2 approved operation mode. ***
```

When not in the approved mode, returned parameters will not include this statement.

[31] A given cryptographic module may only be active on a single domain at a time. In order to transfer between one domain to another, the cryptographic module must be zeroized erasing all CSP before the transfer can occur.