THALES

SafeNet Cryptovisor K7 Cryptographic Module

LEVEL 3 NON-PROPRIETARY SECURITY POLICY

USED AS A STANDALONE DEVICE OR AS THE CRYPTOGRAPHIC MODULE IN SAFENET DATA PROTECTION ON DEMAND

Document Information

Document Part Number	002-010981-001
Release Date	22nd January 2020

Revision History

Revision	Date	Reason
Rev L	11th October 2019	Update to add firmware version 1.3.
Rev. M	24nd January 2020	Update to add firmware version 1.4.

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

Acronyms and abbreviations

Term	Definition
ADL	Authorised 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
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 Cerificate
DoMK	Domain Messaging Keys
DOC	Domain Origin Certificate
DOK	Domain Origin Key

Term	Definition
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
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

Term	Definition
HMAC	Hash-based Message Authentication Code
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
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

Term	Definition
NDRNG	Non-Deterministic Random Number Generator
OAEP	Optimal Asymmetric Encryption Padding
ParEK	Partition Encryption Key
PCI-E	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
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

PREFACE

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

General information on SafeNet 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://safenet.gemalto.com.

> product manuals and technical support literature is available from the Thales Customer Support Portal at https://supportportal.gemalto.com.

> technical or sales representatives of Thales can be contacted through one of the channels listed on https://safenet.gemalto.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 SafeNet Cryptovisor K7 Cryptographic Module.

1.2 Scope

This document applies to hardware versions 808-000048-002 and 808-000073-001 with firmware version 1.1, 1.3 and 1.4, with bootloader version 1.1.1, 1.1.2 and 1.1.4. 808-000048-002 corresponds to a module with fans on the outside of the metal enclosure factory installed. 808-000073-001 corresponds to a module with extra heatsinks installed (instead of fans as pictured).

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

The SafeNet Cryptovisor K7 Cryptographic Module can be used as follows:

> as a standalone device called the SafeNet Cryptovisor K7 Cryptographic Module;

> as an embedded device in the SafeNet Cryptovisor Network HSM.

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
EMI/EMC	3
Self-Tests	3

Security Requirements Section	Level
Design Assurance	3
Mitigation of Other Attacks	3
Cryptographic Module Security Policy	3

1.4 Functional Overview

The SafeNet Cryptovisor K7 Cryptographic Module is a multi-chip embedded cryptographic module in the form of a PCI-Express 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 PCI-E card. "SafeNet Cryptovisor K7 Cryptographic Module cryptographic boundary" on page 14 depicts the SafeNet Cryptovisor K7 Cryptographic Module and "Luna PED and PDA iKey" on page 16 depicts the 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 "Approved Mode of Operation" on page 46 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 PCI-E communications interface. The Trusted Path local PIN Entry Device (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 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 which is available by itself as the SafeNet Cryptovisor K7 Cryptographic Module or embedded within the SafeNet Cryptovisor Network HSM.

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 PCI-E card as outlined. The fans depicted alongside the removable backup battery are not included in the cryptographic boundary.

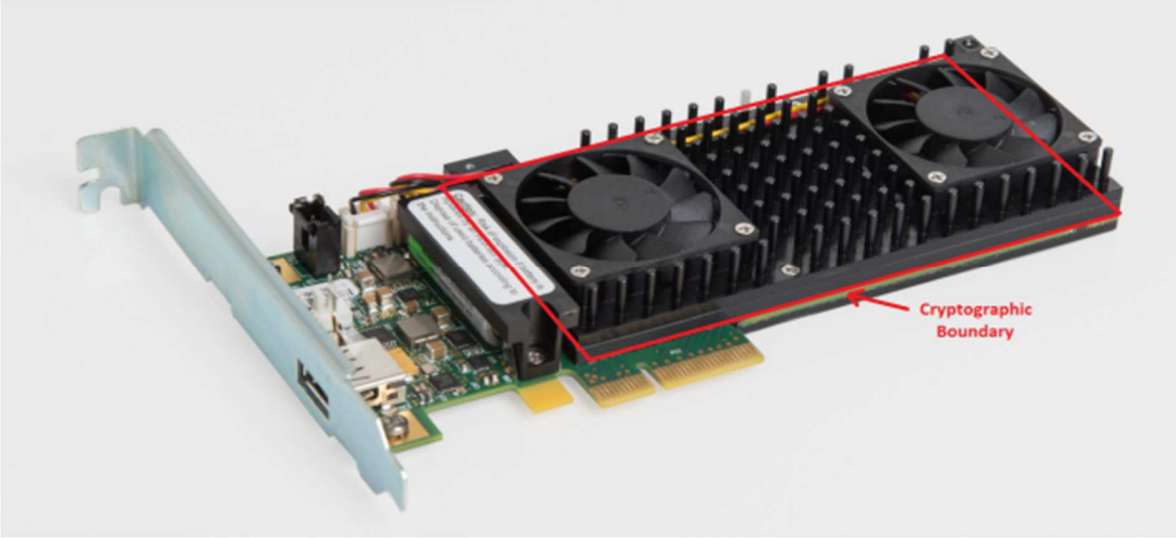

Figure 1: SafeNet Cryptovisor K7 Cryptographic Module cryptographic boundary

2.2 Ports and Interfaces

The module supports the following physical ports and interfaces:

- PCI-E interface
- USB port
- Serial port
- Power supply
- Battery
- LED
- External event input
- 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	PCI-E interface	Data I/O Luna ICD XTC channel for ICD Domain Administration API Bootloader command protocol
	USB	Physical Trusted Path (Local PED)
	Serial interface	Bootloader command protocol
Data Output	PCI-E interface	Data I/O Luna ICD XTC channel for ICD Domain Administration API Bootloader command protocol
	USB	Physical Trusted Path (Local PED)
	Serial Port	Bootloader command protocol
Control Input	PCI-E 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	PCI-E interface	Data I/O Luna ICD XTC channel for ICD Domain Administration API Bootloader command protocol
	USB	Physical Trusted Path (Local 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 PCI-E interface)	N/A
	3.6V battery	N/A

2.3 Trusted Path

The module uses a Luna PED as an external data input/output device in support of authenticating the PDA role. The 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 Luna PED is stored in an iKey[1] USB device connected to the 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 Luna PED:

> Blue (HSM) iKey – for the storage of PDA authentication data2.

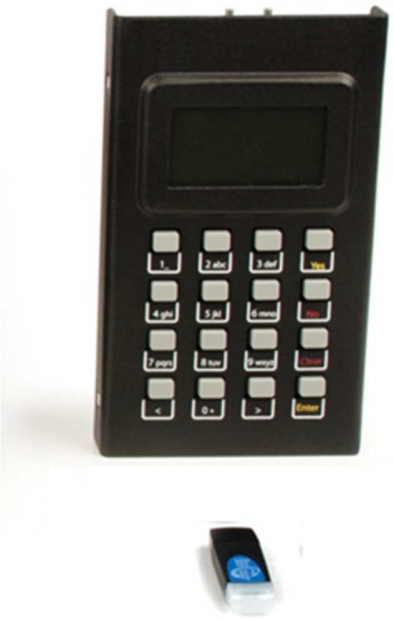

Figure 2: Luna PED and PDA iKey

[1]Within this document the terms "PED key" and "iKey" are interchangeable unless otherwise indicated.
[2]Separate PED Keys are used when M of N token splitting is used to share responsibilities for this role between different operators.

2.4 Secure Messaging

Each user partition uses a secure messaging feature called XTC. An XTC channel is a cryptographic tunnel established between a partition and the SafeNet 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 taking an ephemeral key supplied by the client and a partition's static key and applying ECDH (SP800-56A key agreement). The agreed key is used to transport a tunnel key (using SP800-56B Key Transport). A token and partition secret are then used to derive that tunnel key in the HSM (SP800-108 KDF in counter mode).

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

2.5 Roles and Services

2.5.1 Roles

The SafeNet Cryptovisor K7 Cryptographic Module supports the following roles:

Table 3: SafeNet 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.[3] > Initialize the CVM Role.
Cryptovisor Manager (CVM)	> Domain-level role. > Creates partitions. > Imports partition and session containers to a HSM.

[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 Security Officer (PSO)	> User partition-level role. > Configures container policies for user partition. > Initialize the PCO Role.
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

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

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 "FIPS-Approved Algorithm Implementation" on page 27.

When the module is operating in FIPS 140-2 Approved mode as described in "Approved Mode of Operation" on page 46, the non-Approved Security Functions in "Non-Approved Algorithm Implementations" on page 30 are disabled and cannot be used for these services.

The non-Approved functions in the table 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 "Critical Security Parameters" on page 32.

Table 5: Roles and Access Rights by Service

Service	Critical Security Parameters	Type(s) of Access	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 State.	Use	-	-	-	-	-	x
	PDA Pin (as written to iKey on PDA role creation), SMK, DOK, DOC, DoMK, DoMC, DRBG State.	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	-	-	-	-	-
Download Provider Domain Object (Session Device)	ROOT, HOC, MIC, ECC MIC, ECC HOC, PDA Pin, SMK, JP-PTK, JP(R)-CEK,	Use	x	-	-	-	-	-

Service	Critical Security Parameters	Type(s) of Access	Role					
			PDA	CVM	PSO	PCO	PCU	PU
	JP(T)-CEK.	Write						
Transfer Provider Domain Object (Root Device)	ROOT, HOC, MIC, ECC MIC, ECC HOC, JP(T)-CEK, JP-PTK	Use	x	-	-	-	-	-
Activate a device	PDA Pin.	Use	x	-	-	-	-	-
Deactivate a Device	N/A	N/A	x	-	-	-	-	-
Login	Authentication data, PEC/PEK, DRBG State.	Use	x	x	x	x	x	-
	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 State.	Use	-	x	-	-	-	-
	USK, PSK, XTC-PMK, XTC-PMC, PFK, XTC-PToK.	Write						
Delete Partition	CVM Password	Use	-	x	-	-	-	-
Initialize Partition	USK	Use	-	-	x	-	-	-
Configure Partition Policy	N/A	N/A	-	-	x	-	-	-
Open XTC Session	XTC-epCK, XTC-PMK, XTC-PMC, XTC-PT, XTC-TDK, XTC-TTC,	Use	-	-	x	x	x	x
	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

Service	Critical Security Parameters	Type(s) of Access	Role					
			PDA	CVM	PSO	PCO	PCU	PU
Create Partition User	PSO Admin Password (for CO) or CO Password (for CU), PSK, USK, DRBG State.	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	-	-	-	-	-
Key Generation	DRBG State, USK[5].	Use	-	-	-	x	-	-
	Symmetric keys (general partition or session keys).	Write						
Key Pair Generation	DRBG State, USK[6].	Use	-	-	-	x	-	-
	Asymmetric key pairs (general partition or session keys).	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).

Service	Critical Security Parameters	Type(s) of Access	Role					
			PDA	CVM	PSO	PCO	PCU	PU
Wrap Symmetric Key	USK[7], DRBG State, KTS symmetric/asymmetric wrapping key.	Use	-	-	-	x	-	-
Wrap Asymmetric Key	USK[8], DRBG State, KTS symmetric wrapping key.	Use	-	-	-	x	-	-
Unwrap Symmetric/Asymmetric Key	KTS symmetric unwrapping key, symmetric unwrapping key.	Use	-	-	-	x	-	-
	Symmetric or Asymmetric unwrapped key.	Write						
Encrypt/Decrypt (Symmetric Algorithm)	USK[9], DRBG State, Symmetric keys (general partition or session keys)	Use	-	-	-	x	x	-
Encrypt/Decrypt (Asymmetric Algorithm)	USK[10], DRBG State, Asymmetric keys (general partition or session keys)	Use	-	-	-	x	x	-
Signature Generation (Public Key Cryptography)	USK[11], DRBG State, RSA, DSA, ECDSA private keys (general partition or session keys)	Use	-	-	-	x	x	-
Signature Verification (Public Key Cryptography)	RSA, DSA, ECDSA public keys (general partition or session keys)	Use	-	-	-	x	x	-
Generate Hash Value	N/A	N/A	-	-	-	x	x	-

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

Service	Critical Security Parameters	Type(s) of Access	Role					
			PDA	CVM	PSO	PCO	PCU	PU
MAC Generation	USK[12], DRBG State, Symmetric keys (general partition or session keys)	Use	-	-	-	x	x	-
MAC Verification	USK[13], DRBG State, Symmetric keys (general partition or session keys)	Use	-	-	-	x	x	-
Key Derivation	USK[14], DRBG State, Symmetric keys (general partition or session keys), Asymmetric keys (general partition or session keys).	Use	-	-	-	x	-	-
	Symmetric keys (general partition or session keys)	Write						
Retrieve DRBG output for export.	DRBG State	Use	-	-	-	x	x	x
Store Data Object	Non-cryptographic data	Write	-	-	-	x	x	-
Read Data Object	Non-cryptographic data	Use	-	-	-	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

[12] Where symmetric keys used are “token” rather than “session” objects.
[13] Where symmetric keys used are “token” rather than “session” objects.
[14] Where symmetric or private keys used are “token” rather than “session” objects.

2.6 Authentication

All roles except for the Public User must authenticate to the module by providing their authentication data. "Roles and Required Identification and Authentication" on page 24 and "Strengths of Authentication Mechanisms" on page 24 explains the type and strength of the authentication data supported for each role.

All roles must authenticate using either presentation of a PED 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 to the local PED.

Table 6: Roles and Required Identification and Authentication

Role	Type of Authentication	Authentication Data
Provider Domain Administrator	Identity-based	Authentication token (PED Key) with optional PIN.
Cryptovisor Manager	Identity-based	Password
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
PED Key (PDA)	Authentication is based on presentation of a 48 byte random value generated when a role is initialized and stored on a PED key. The probability of guessing the authentication data in a single attempt is 1 in 2^{384}. With a maximum of 6000 failed login attempts per minute, 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 failed login attempts per minute, the thresholds required by FIPS 140-2 can never be reached.

Authentication Mechanism	Strength of Mechanism
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 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 Luna PED in order to reconstruct the original secret.

2.7 Physical Security

The SafeNet Cryptovisor K7 Cryptographic Module is a multi-chip embedded module as defined by FIPS PUB 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. 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 PCI-E Card Removal

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

Note, under-voltage conditions cannot be reliably 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 in to STM, a random string and a fingerprint of the internal state of the module is output from the module. The fingerprint is a SHA-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[15] 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.

[15] 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.

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 FIPS-Approved Algorithm Implementations

The FIPS-Approved algorithms implemented by the module are listed in the table below:

Table 8: FIPS-Approved Algorithm Implementation

Approved Security Functions	Certificate No.
Symmetric Encryption/Decryption	
AES: ECB, CBC, OFB, CTR, CFB8, CFB128, GCM[16], XTS, KW, KWP (128, 192, 256-bits)	AES #5652
AES: GCM[17] (128, 192, 256-bits)	AES #5653
Hashing	
SHA: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (Byte Only)	SHS #4533
SHA SHA-256, SHA-512 (Byte Only)	SHS #4534
SHA: SHA-1, SHA-384 (Byte Only)	SHS #3951 and SHS #3952
Message Authentication Code	
HMAC: HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	HMAC #3766
AES: CMAC(128, 192, 256-bits)	AES #5652
Asymmetric	

[16] The module generates IVs internally using the Approved DRBG which are at least 96-bits in length.
[17] The module generates IVs internally using the Approved DRBG which are at least 96-bits in length.

RSA: Key Generation (2048 and 3072 modulus), Signature Generation (1024-4096 modulus), Signature Verification (1024-4096 modulus)	RSA #3042
RSA: Key Generation (2048 and 3072 modulus)	RSA #3043
RSA: Signature Verification (4096 modulus)	RSA #2631 and RSA #2632
DSA: Parameter Generation (2048 and 3072 modulus), Key Generation (2048 and 3072 modulus), Signature Generation (2048 and 3072 modulus), Signature Verification (1024-3072 modulus)	DSA #1452
DSA: Parameter Generation (2048 and 3072 modulus), Key Generation (2048 and 3072 modulus)	DSA #1453
ECDSA: Key Generation, Signature Generation, Signature Verification 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	ECDSA #1526
ECDSA: Key Generation 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	ECDSA #1527
ECDSA (CVL): Signature Generation Component 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	CVL #2047
RSA (CVL): Decryption Primitive	CVL #2043 and CVL #2044
Key Agreement Scheme	
ECC: Ephemeral Unified, OnePassDH FFC: dhHybrid1, dhEphem, dhHbryidOneFlow, dhOneFlow	KAS #195
FFC: dhHybrid1, dhEphem, dhHbryidOneFlow, dhOneFlow	KAS #196
Key Transport	

KTS (AES Cert. #5652) (128, 192, 256-bits)	AES #5652
Key Derivation Function	
Key-Based Key Derivation Function (KBKDF) CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512 : Counter Mode	KDF #234
Random Number Generation	
NIST SP 800-90A DRBG (CTR) AES 256	DRBG #2283
Key Generation	
CKG[18]	Vendor Affirmed

Table 9: Allowed Security Function for the Firmware Implementation

Allowed Security Functions
Key Agreement
Diffie-Hellman (key agreement; key establishment methodology provides 112 bits of encryption strength)
EC Diffie-Hellman (key agreement; key establishment methodology provides 192 bits of encryption strength)
Key Transport
RSA (CVL Certs. #2043 and #2044, key wrapping; key establishment methodology provides between 112 and 150 bits of encryption strength)
AES (key unwrapping; key establishment methodology provides between 128 and 256 bits of encryption strength) (based on AES Cert. #5652)
Entropy Source
NDRNG[19]

[18] Resulting symmetric keys and seeds used for asymmetric key generation are an unmodified output from an Approved DRBG.
[19] the entropy source falls within a scenario of IG 7.14 that requires an entropy assessment and meets requirements of IG 7.15.

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 "Identifying the Module Version" on page 45.

Table 10: Non-FIPS Approved Security Functions

Non-FIPS Approved Security Functions
Symmetric Encryption/Decryption
DES
RC2
RC4
RC5
CAST3
CAST5
SEED
ARIA
Hashing
MD2
MD5
HAS-160
SM3
Message Authentication Code
AES-MAC
DES-MAC
RC2-MAC
RC5-MAC
CAST3-MAC
CAST5-MAC
SEED-MAC
ARIA-MAC

Non-FIPS Approved Security Functions
SSL3-MD5-MAC
SSL3-SHA1-MAC
HMAC (non-compliant for any configuration providing less than 112 bits of security strength)
Asymmetric
KCDSA
RSA X-509
RSA (non-compliant less than 112 bits of security strength)
DSA (non-compliant less than 112 bits of security strength)
ECDSA (non-compliant less than 112 bits of security strength)
Deterministic ECDSA
EdDSA
Key Generation
DES
RC2
RC4
RC5
CAST3
CAST5
SEED
ARIA
GENERIC-SECRET
Key Agreement
ECC (non-compliant less than 112 bits of encryption strength)
Diffie-Hellman (key agreement; key establishment methodology; non-compliant less than 112 bits of encryption strength)
Key Transport

Non-FIPS Approved Security Functions
RSA (key wrapping; key establishment methodology; non-compliant 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	Input from host using Luna ICD communication path over XTC.	User provided password input by the operator as authentication data.
PED Authentication Data	48-byte random value	NIST SP 800-90A DRBG	Input / output via direct connection to PED.	Random value that is generated by the module when a role is created and is written out to the PED key via the Trusted Path.
Password (Authentication Data) – CVM	16-byte	NIST SP 800-90A DRBG	Copy output on creation encapsulated in a JWE encrypted package under a client generated CEK.	A 16-byte random value that is base64 encoded and returned to the PDA on creation of a Cryptovisor Manager user.
Key Cloning Domain Vector (KCV)	48-byte value	Derived from password using concatenation KDF consistent with SP800-108	Password entered by user over XTC.	Value that is used to control a partitions ability to participate in the cloning protocol. It is input by the operator at the time the module or partition is initialized.
User Storage Key (USK)	AES-KWP 256	NIST SP 800-90A DRBG	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-KWP 256	NIST SP 800-90A DRBG	Not input or output.	This key is used to encrypt all sensitive attributes of all private objects owned by the PDA and stored in the Admin Partition.
Partition Storage Key (PSK)	AES-GCM 256	NIST SP 800-90A DRBG	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-KWP 256	NIST SP 800-90A DRBG	Not input or output.	AES key that is the same for all users on a specific Cryptovisor cryptographic 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	Loaded at manufacturing	Certificate embedded in 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, Firmware and License Signing Certificates (MIC, FSC and LSC).
Manufacturer's Integrity Certificate (MIC)	RSA-4096 public key certificate	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).
ECC Manufacturer's Integrity Certificate (ECC MIC)	EC-secp384r1 public key certificate	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). It is self-signed. Used in verifying ECC HOC.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Hardware Origin Key (HOK)	RSA 4096 bit private key	FIPS PUB 186-4, Appendix B.3.6.	Not input or output.	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 SHA-384) It is generated at the time the device is manufactured.
Hardware Origin Certificate (HOC)	RSA-4096 public key certificate	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 PUB 186-4, Appendix B.4.1.	Not input or output.	ECC 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	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)
Password Encryption Key (PEK)	RSA 4096 bit private key	FIPS PUB 186-4, Appendix B.3.6.	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 isn't persistent over a power-cycle.
Password Encryption Certificate (PEC)	RSA-4096 public key certificate	FIPS PUB 186-4, Appendix B.3.6.	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.
Token Unwrapping Key 3 (TUK3)	RSA 2048 bit private key	FIPS PUB 186-4, Appendix B.3.6.	Not input or output.	RSA private key used to transfer source and target nonce as part of the cloning protocol.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Token Wrapping Key 3 (TWC3)	RSA-2048 bit public key certificate	FIPS PUB 186-4, Appendix B.3.6.	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 the cloning protocol.
Firmware Signing Cert (FSC)	RSA-4096 bit public key certificate.	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 Gemalto 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.	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 Gemalto Root signing key. Used to verify CUF on load.
Manufacturer Authentication Certificate (MAC)	RSA-2048 bit public key certificate	FIPS PUB 186-4, Appendix B.3.6.	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	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 PUB 186-4, Appendix B.3.6.	Not input or output.	RSA private key used for a specific PKI implementation requiring assurance that a key or a specific action originated within the hardware crypto module.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Device Authentication Certification (DAC)	RSA-2048 public key certificate	FIPS PUB 186-4, Appendix B.3.6.	Certificate output in plaintext.	The X.509 public key certificate corresponding to the DAK. It is signed by the MAK (private key corresponding to MAC). 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 PUB 186-4, Appendix B.4.1.	Not input or output.	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 ECC MAC).
ECC Device Authentication Certificate (ECC DAC)	EC secp384r1 public key certificate	FIPS PUB 186-4, Appendix B.4.1.	Not input or output.	The X.509 public key certificate corresponding to the ECC DAK. It is signed by the ECC HOK.
Key Encryption Key (KEK)	AES-KWP 256	NIST SP 800-90A DRBG	Not input or output.	The KEK encrypts all sensitive values and is zeroized in response to a decommission signal.
DRBG Key	AES-256	Hardware Random Source	Not input or output.	AES key stored in the RAM. Used in an implementation of the NIST SP 800-90A CTR (AES) DRBG.
DRBG Seed	384 bits	Hardware Random Source	Not input or output.	Random seed data generated from conditioned output from the module NDRNG and used in the implementation of the NIST SP 800-90A CTR (AES) DRBG.
DRBG V	128 bits	Hardware Random Source	Not input or output.	Part of the secret state of the approved DRBG. The value is generated using the methods described in NIST SP 800-90A.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
DRBG Entropy Input	384 bits	Hardware Random Source	Not input or output.	Entropy value used to initialize the approved DRBG and taken from the conditioned output from the module NDRNG.
Domain Origin Key (DOK)	RSA-4096	FIPS PUB 186-4, Appendix B.3.6.	Input/output using AES-GCM as part of PDO. Encrypted under JP-PTK.	Key signing for domain and partition key hierarchy for origin and messaging certificates. Used to authenticate messages from HSM to the client.
Domain Origin Cert (DOC)	RSA-4096	FIPS PUB 186-4, Appendix B.3.6.	Certificate output in plaintext.	Used to authenticate the Domain Messaging Key alongside XTC – Partition Messaging Key from HSM to the client. The DOC is signed with the HOK of the HSM used to create the first instance of a PDO. Clients can validate the authenticity of the DOC based on supplied certificate chain including the HOC which chains back to the shared root.
Domain Messaging Key (DoMK)	RSA-4096	FIPS PUB 186-4, Appendix B.3.6.	Input/output using AES-GCM as part of PDO. Encrypted under JP-PTK.	Secure communication between client and HSM domain. RSA-OAEP (Basic from SP 800-56B) with SHA1 MGF. Messages are decrypted with the DoMK.
Domain Messaging Cert (DoMC)	RSA-4096	Generated by the HSM based on the Domain Messaging Key.	AES-GCM encrypted under JOSE response key.	Secure communication between client and HSM domain. RSA-OAEP with SHA1 MGF. JOSE-CEK is encrypted with this key.
Device Messaging Key (DeMK)	RSA-4096	FIPS PUB 186-4, Appendix B.3.6.	Not input or output.	Used for JOSE/JWE messaging API to communicate between client and HSM. RSA-OAEP with SHA1 MGF and SHA256 MFG options.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
Device Messaging Cert (DeMC)	RSA-4096	RSA-OEAP (JWE) encryption on client	Certificate output in plaintext.	Verified by the client. Used for JOSE/JWE messaging API to communicate between client and HSM. RSA-OAEP with SHA1 MGF
Domain Encryption Master Key (DEK)	AES-GCM 256	NIST SP 800-90A DRBG	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
Join-Protocol (Target) Content Encryption Key (JP(T)-CEK)	AES-GCM 256	NIST SP 800-90A DRBG	Input/output between HSMs. RSA-OAEP SHA384 MGF.	Used to encrypt and authenticate join request messages (AES-GCM).
Join-Protocol - PDO Transfer Key (JP-PTK)	AES-GCM 256	Generated by Target Device in join-protocol using CTR-DRBG with AES256 cipher.	Exported from target device for Join protocol encrypted 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-GCM 256	NIST SP 800-90A DRBG	Input/output between HSMs. RSA-OAEP SHA384 MGF.	Used to encrypt and authenticate join response messages (AES-GCM).
Partition Export Key (ParEK)	AES-GCM 256	Derived from Domain Encryption Master Key using SP800-108 KDF.	Input/output using AES-GCM as part of PDO. Encrypted under JP-PTK.	Used to export partition.
Partition Fragment Key (PFK)	AES-GCM 256	NIST SP 800-90A DRBG	Not input or output.	This is key is to encrypt partition objects and sessions for a given partition.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
XTC Partition Messaging Key (XTC-PMK)	EC secp384r1 private key	FIPS PUB 186-4, Appendix B.4.1.	Input/output using AES-GCM under the Partition Export Key.	The HSMs static private key used in EC-DH negotiation of the XTC tunnel transport key.
XTC Partition Messaging Certificate (XTC-PMC)	EC secp384r1 public key certificate	FIPS PUB 186-4, Appendix B.4.1.	AES-GCM encrypted under JOSE response key.	The HSMs static public key used in EC-DH negotiation of the XTC tunnel transport key.
XTC ephemeral client key (XTC-epCK)	EC secp384r1 public key	Client generated	Input using AES-GCM encrypted under JOSE Content Encryption Key.	The clients ephemeral key pair used in EC-DH negotiation of the XTC tunnel transport key.
XTC tunnel transport key (XTC-TTK)	AES-GCM 256	ECDH (1e,1s) from SP800-56A as base then derived using SP800-108 KDF.	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	NIST SP 800-90A DRBG	Input/output using AES-GCM encrypted under the Partition Export Key.	Used a secret in key derivation of XTC partition tunnel key.
XTC Partition Tunnel Key (XTC-PTuK)	AES-GCM 256	SP 800-56A Rev. 3 section 5.8.1 KDF	Transmitted to client using XTC tunnel transport key.	Used to protect Luna ICD commands between the SafeNet Client and cryptographic module.
XTC Partition Token Key (XTC-PToK)	AES-GCM 256	NIST SP 800-90A DRBG	Input/output using AES-GCM encrypted under the Partition Export Key.	Used to sign XTC partition token.
XTC Partition Token (XTC-PT)	Token (blob)	NIST SP 800-90A DRBG 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	NIST SP 800-90A DRBG	Transmitted encrypted using AES-GCM as part of XTC tunnel under XTC Partition Tunnel Key.	Used to prove access to a session group.

CSP Name	CSP Type	Generation Method	Input / Output Method	Description
JOSE response key (JOSE-RK)	AES-GCM 256	Client generated	Input encrypted using RSA-OAEP 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-GCM 256	Client generated	Input with RSA-OAEP (SHA1 MGF). Not Output	Encrypts responses to JOSE/JWE API as used for the Domain Administration API.
Asymmetric Key Pairs (general partition or session keys)	RSA, DSA, ECC, DH	N/A (user imported) or NIST SP 800-90A DRBG (module generated)	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	N/A (user imported) or NIST SP 800-90A DRBG (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.

2.10.1 Key Generation

Symmetric cryptographic keys are generated by the direct unmodified output of the module's NIST SP 800-90A DRBG. The DRBG output is also used as a seed for asymmetric key generation.

Keys which 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 NIST SP 800-90A.

The NIST SP800-90A DRBG (CTR-DRBG using AES256) is seeded using 2048 bits or raw entropy taken from the module NDRNG which is conditioned using SHA-512 ahead of creating the 384 bit seed. Based on calculated min-entropy values for the platform raw noise source and factors outlined in NIST SP800-90B, the 384-bit input used to seed to the DRBG has a full 384 bits of entropy.

2.10.2 Key Import and Export

Import and Export of CSP is supported over the following interface:

- Physical Trusted Path (local PED) - PED Authentication Data.
- XTC Channel for ICD, logical interface - Password (Authentication Data - PSO, PCO, PCU), KCV, PEC, TWC3, MAC, ECC MAC, DAC, ECC DAC, XTC-SA, Symmetric Keys (general partition or session keys).
- Luna ICD, logical interface - Firmware Signing Cert, Licensing Signing Cert, HOC, ECC HOC, XTC-PToK, XTC-PTuK, XTC-TDK,.XTC-epCK.
- Domain Administration API, logical interface - Password (Authentication Data) – CVM, DOC, DoMC, DeMC, DEK,PEK, USK, PSK, SMK, JP(T)-CEK, JP-PTK, JP(R)-CEK, ParEK, XTC-PMC, XTC-PMK, XTC-PMC, JOSE-CEK, JOSE-RK.

For details of encryption and specific CSP used refer to Input/Output column of "Summary of CSPs" on page 32.

The remaining keys and CSPs listed in "Summary of CSPs" on page 32 and not listed above against an interface are not input to or output from the module.

Depending on the configuration of the module, the following methods of key entry and output are available as a service (see "Services" on page 19):

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

  Key cloning is a product feature that uses AES-256 in CBC mode with a single-use key to encrypt an object being transferred from one user partition to another user partition accessible to a Partition Crypto Officer. These partitions can be on the same or different cryptographic modules. Objects transferred using the cloning protocol may be keys, user data, or module data. The single-use AES key is obtained by combining the 48 byte key cloning domain vector (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 4096-based key transport. The ‘One-Step Key Derivation’ function from SP800-56C is used with SHA-512 as the auxiliary function.

  All ICD messages involved in the transfer using Key Cloning 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 SP800-56A. Use of AES-GCM provides compliance with SP800-38F.

- **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 (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 SP800-56A. Use of AES-GCM provides compliance with SP800-38F.

> **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 and exported partitions can only be inserted onto an HSM containing a copy of the PDO used for the domain the partitions were exported from.

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, and 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 12: Power On Self Tests (Bootloader) – Module Integrity

Test	When Performed	Indicator
Boot loader performs an RSA 4096-bit SHA-384 signature verification of itself	Power-on	Error output and module halt

Test	When Performed	Indicator
Boot loader performs an RSA 4096-bit SHA-384 signature verification of the firmware prior to firmware start	Power-on/Request[20]	Error output and module halt

Table 13: 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	Error output and module halt
SHA KAT (SHA1, SHA-224,SHA256, SHA-384, SHA-512)	Power-on/Request	Error output and module halt
HMAC KAT (HMAC-SHA1, HMAC-SHA224, HMAC-SHA-256, HMAC SHA-384, HMAC SHA-512)	Power-on/Request	Error output and module halt
RSA KAT (Signature Generation, Sig Verification)	Power-on/Request	Error output and module halt
DSA KAT (Signature Generation, Sig Verification)	Power-on/Request	Error output and module halt
Diffie-Hellman KAT (X9.42 DH Derive)	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).	Power-on/Request	Error output and module halt.
ECDH KAT (Derive)	Power-on/Request	Error output and module halt
ECDSA KAT (Signature Generation, Sig Verification)	Power-on/Request	Error output and module halt.
KDF KAT (Derive using AES-CMAC, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512 as PRF).	Power-on/Request	Error output and module halt.

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.

[20] Request indicates triggering a POST via a command

Table 14: Conditional Self Tests

Test	When Performed	Where Performed	Indicator
NDRNG conditional tests[21]	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
ECDSA – Pair-wise consistency test (asymmetric key pairs)	On generation	Firmware	Error output
Firmware load test (4096-bit RSA sig ver)	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

[21] 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 IG 9.8 for modules implementing an approved DRBG from NIST SP800-90A.

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 "Scope" on page 12. 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 1.1.2.
```

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: 1.1.0
Hardware serial number: 551982
Device time: Mon Jul 30 13:53:52 2018 (UTC) and 187992 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 -> 64792103551
Model -> Cryptovisor K7
Firmware Version -> 7.1.3
CV Firmware Version -> 1.1.0
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 '7.1.3' 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 a valid number for a module with hardware compliant with this security policy is shown below:

```
HSM Part Number -> 808-000069-001
```

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 "Scope" on page 12.

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[22] – 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 – this can separately be done by calling `partition showinfo` 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.

[22] 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.