Non-Proprietary Security Policy

NITROX III CNN35XX-NFBE HSM Family

Document number:	CNN35xx-NFBE-SPD-L3
Document Version:	Version 2.09-0702
Revision Date:	05/08/20244

This document may be reproduced only in its original entirety [without revision].

Revision History

Revision	Date	Author	Description of Change
2.08.01	04/21/2021	Girish Kumar Yerra Rajendar Kalwa	FW 2.08 build 09 CMVP Submission.
2.08.02	11/11/2021	Girish Kumar Yerra Rajendar Kalwa	FW 2.08 build 10 CMVP Submission updates.
2.08.03	06/02/2022	Girish Kumar Yerra Rajendar Kalwa	Addressed comments from NIST. Updated FW build with bug fixes to 2.08 build 11.
2.08.04	02/10/2023	Rajendar Kalwa	Addressed comments from NIST. Updated FW build with bug fixes to 2.08 build 12.
2.09.07	08/10/2023	Rajendar Kalwa	Addressed comments from NIST. Updated FW build with changes to address CMVP comments and bug fixes to 2.09 build 07.
2.09.0701	12/08/2023	Rajendar Kalwa Vikash Kumar	Addressed comments from NIST. Updated FW build to exclude Transition algorithms which were allowed as per IG D.G.
2.09.0702	05/08/2024	Rajendar Kalwa Vikash Kumar	Addressed comments from NIST.

Contents

List of Tables

List of Figures

1 General

1.1 Security Level

The cryptographic module meets the overall requirements applicable to Level 3 security of FIPS 140-3.

Table 1 – Security Levels

ISO/IEC 24759 Section 6 [Number Below]	FIPS 140-3 Section Title	Security Level
1	General	3
2	Cryptographic Module Specification	3
3	Cryptographic module interfaces	3
4	Roles, Services and Authentication	3
5	Software/Firmware Security	3
6	Operational Environment	N/A
7	Physical Security	3
8	Non-invasive security	N/A
9	Sensitive Security parameter management	3
10	Self-tests	3
11	Life-cycle assurance	3
12	Mitigation of Other Attacks	N/A

2 Cryptographic Module Specification

2.1 Module Overview

The NITROXIII CNN35XX-NFBE HSM Family module (hereafter referred to as *the module or HSM*) by Marvell is a high-performance purpose-built security solution for crypto acceleration. The module provides a FIPS 140-3 overall Level 3 security solution. The module is deployed in a PCIe slot to provide crypto and TLS 1.0/1.1/1.2 acceleration in a secure manner to the system host. It is typically deployed in a server or an appliance to provide crypto offload. The module's functions are accessed over the PCIe interface via opcodes defined by the module.

The module is a hardware multi-chip embedded cryptographic module. The module provides cryptographic primitives to accelerate approved and allowed algorithms for TLS 1.0/1.1/1.2 and SSH. The cryptographic functionality includes modular exponentiation, random number generation, and hash processing, along with protocol specific complex instructions to support TLS 1.0/1.1/1.2 security protocols using the embedded NITROXIII chip. The module implements password based single factor authentication at FIPS 140-3 Level 3 security. The physical boundary of the module is the outer perimeter of the PCIe card itself, as depicted in section 2.13.

The configuration of hardware and firmware for this validation is:

Table 2 – Hardware Part Numbers

Part Number	HW Version	LiquidSecurity Appliance	Cores Enabled	Key Store Size	Max Partitions
CNL3560P-NFBE-G	HW-1.0	Yes	64	100K	32
CNL3560-NFBE-G	HW-1.0	Yes	64	100K	32
CNL3530-NFBE-G	HW-1.0	Yes	32	25K	32
CNL3510-NFBE-G	HW-1.0	Yes	24	25K	24
CNL3510P-NFBE-G	HW-1.0	Yes	32	50K	32
CNL3560P-NFBE-2.0-G	HW-2.0	Yes	64	100K	32
CNL3560-NFBE-2.0-G	HW-2.0	Yes	64	100K	32
CNL3530-NFBE-2.0-G	HW-2.0	Yes	32	25K	32
CNL3510-NFBE-2.0-G	HW-2.0	Yes	24	25K	24
CNL3510P-NFBE-2.0-G	HW-2.0	Yes	32	50K	32
CNL3560PB-NFBE-2.0-G	HW-2.0	Yes	64	100K	32
CNL3560B-NFBE-2.0-G	HW-2.0	Yes	64	100K	32
CNL3530B-NFBE-2.0-G	HW-2.0	Yes	32	25K	32
CNL3510B-NFBE-2.0-G	HW-2.0	Yes	24	25K	24
CNL3510PB-NFBE-2.0-G	HW-2.0	Yes	32	50K	32
CNL3560P-NFBE-3.0-G	HW-3.0	Yes	64	100K	32
CNL3560B-NFBE-3.0-G	HW-3.0	Yes	64	100K	32
CNL3560B-NFBE-3.0-G-FB	HW-3.0	Yes	64	100K	32
CNL3560-NFBE-3.0-G	HW-3.0	Yes	64	100K	32
CNL3560A-NFBE-3.0-G	HW-3.0	Yes	64	100K	32
CNL3560C-NFBE-3.0-G	HW-3.0	Yes	64	100K	32
CNL3560D-NFBE-3.0-G	HW-3.0	Yes	64	100K	32
CNL3560E-NFBE-3.0-G	HW-3.0	Yes	64	100K	32
CNL3560F-NFBE-3.0-G	HW-3.0	Yes	64	100K	32
CNL3530-NFBE-3.0-G	HW-3.0	Yes	32	25K	32
CNL3530B-NFBE-3.0-G	HW-3.0	Yes	32	25K	32
CNL3530A-NFBE-3.0-G	HW-3.0	Yes	32	25K	32
CNL3530C-NFBE-3.0-G	HW-3.0	Yes	32	25K	32
CNL3530D-NFBE-3.0-G	HW-3.0	Yes	32	25K	32
CNL3530E-NFBE-3.0-G	HW-3.0	Yes	32	25K	32
CNL3530F-NFBE-3.0-G	HW-3.0	Yes	32	25K	32

Part Number	HW Version	LiquidSecurity Appliance	Cores Enabled	Key Store Size	Max Partitions
CNL3510-NFBE-3.0-G	HW-3.0	Yes	24	25K	24
CNL3510P-NFBE-3.0-G	HW-3.0	Yes	32	50K	32
CNL3510A-NFBE-3.0-G	HW-3.0	Yes	32	50K	32
CNL3510C-NFBE-3.0-G	HW-3.0	Yes	32	50K	32
CNL3510D-NFBE-3.0-G	HW-3.0	Yes	32	50K	32
CNL3510E-NFBE-3.0-G	HW-3.0	Yes	32	50K	32
CNL3510F-NFBE-3.0-G	HW-3.0	Yes	32	50K	32
CNL3510I-NFBE-3.0-G	HW-3.0	Yes	24	25K	16
CNN3560P-NFBE-G	HW-1.0	No	64	100K	64
CNN3560-NFBE-G	HW-1.0	No	64	100K	32
CNN3530-NFBE-G	HW-1.0	No	32	25K	32
CNN3510-NFBE-G	HW-1.0	No	24	25K	24
CNN3510LP-NFBE-2.0-G	HW-2.0	No	24	25K	24
CNN3510LPB-NFBE-2.0-G	HW-2.0	No	24	25K	24
CNN3560P-NFBE-2.0-G	HW-2.0	No	64	100K	64
CNN3560-NFBE-2.0-G	HW-2.0	No	64	50K	32
CNN3530-NFBE-2.0-G	HW-2.0	No	32	25K	32
CNN3510-NFBE-2.0-G	HW-2.0	No	24	25K	24
CNN3505LP-NFBE-2.0-G	HW-2.0	No	16	10K	16
CNN3560P-NFBE-3.0-G	HW-3.0	No	64	100K	64
CNN3560-NFBE-3.0-G	HW-3.0	No	64	50K	32
CNN3560A-NFBE-3.0-G	HW-3.0	No	64	50K	32
CNN3560C-NFBE-3.0-G	HW-3.0	No	64	50K	32
CNN3560D-NFBE-3.0-G	HW-3.0	No	64	50K	32
CNN3560E-NFBE-3.0-G	HW-3.0	No	64	50K	32
CNN3560F-NFBE-3.0-G	HW-3.0	No	64	50K	32
CNN3530-NFBE-3.0-G	HW-3.0	No	32	25K	32
CNN3530A-NFBE-3.0-G	HW-3.0	No	32	25K	32
CNN3530C-NFBE-3.0-G	HW-3.0	No	32	25K	32
CNN3530D-NFBE-3.0-G	HW-3.0	No	32	25K	32
CNN3530E-NFBE-3.0-G	HW-3.0	No	32	25K	32
CNN3530F-NFBE-3.0-G	HW-3.0	No	32	25K	32
CNN3510-NFBE-3.0-G	HW-3.0	No	24	25K	24

Part Number	HW Version	LiquidSecurity Appliance	Cores Enabled	Key Store Size	Max Partitions
CNN3510A-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510C-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510D-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510E-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510F-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510LP-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510LPB-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510LPA-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510LPC-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510LPD-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510LPE-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3510LPF-NFBE-3.0-G	HW-3.0	No	24	25K	24
CNN3505LP-NFBE-3.0-G	HW-3.0	No	16	10K	16
CNN3505LPA-NFBE-3.0-G	HW-3.0	No	16	10K	16
CNN3505LPC-NFBE-3.0-G	HW-3.0	No	16	10K	16
CNN3505LPD-NFBE-3.0-G	HW-3.0	No	16	10K	16
CNN3505LPE-NFBE-3.0-G	HW-3.0	No	16	10K	16
CNN3505LPF-NFBE-3.0-G	HW-3.0	No	16	10K	16

LP is a low-frequency part, where Nitrox III chip runs at 500MHz; otherwise, it runs at 600MHz.

The LiquidSecurity Appliance is a network accessible, remote configurable, multi-tenant server platform to enable private and Hybrid cloud deployments of NITROX III CNN35XX-NFBE HSM. The LiquidSecurity Appliance is outside the module’s cryptographic boundary and therefore out of scope of this validation.

CNN35XX-NFBE-G Firmware:

CNN35XX-NFBE-FW-2.09-0702

CNN35XX-NFBE-G Secure Machine:

CNN35XX-NFBE-SMW-2.09-0702

CNN35XX-NFBE-G Bootloader:

CNN35XX-UBOOT-4.03-03

Note: These binaries do not have extensions.

The module is considered to be operating in the approved mode only when it is running the firmware/bootloader versions listed above.

The module supports different performance options as listed above in the hardware identifier. The physical hardware and firmware are identical across all options. The underlying hardware has multiple identical cryptographic engines which are enabled or disabled using an option parameter set at manufacturing time. During Manufacturing, the number of cryptographic cores, Key stores and Partitions are enabled by configuration for different Part number. Please refer to Table 2 – Hardware Part Numbers for further details. CNL and CNN part numbers employ the same hardware and firmware; the only difference is the vendor configuration, where CNL part numbers have certificate authentication enabled during manufacturing via the CN_INIT_TOKEN service.

The major blocks of the module are: General purpose MIPS-based control processor, crypto processors, RAM memory, NOR and eMMC flash for persistent storage, USB interfaces, and PCIe gen-2 x8 interfaces.

There are no excluded components within the Module.

Table 3 – Cryptographic Module Specification

Model	Hardware Part Number (Version)	Firmware version	Distinguishing Features
CNN35XX-NFBE	CNL3560P-NFBE-G (HW-1.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	No SMBus and no RTC
CNN35XX-NFBE	CNL3560-NFBE-G (HW-1.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	No SMBus and no RTC
CNN35XX-NFBE	CNL3530-NFBE-G (HW-1.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	No SMBus and no RTC
CNN35XX-NFBE	CNL3510-NFBE-G (HW-1.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	No SMBus and no RTC
CNN35XX-NFBE	CNL3510P-NFBE-G (HW-1.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	No SMBus and no RTC
CNN35XX-NFBE	CNN3560P-NFBE-G (HW-1.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	No SMBus and no RTC
CNN35XX-NFBE	CNN3560-NFBE-G (HW-1.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	No SMBus and no RTC
CNN35XX-NFBE	CNN3530-NFBE-G (HW-1.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	No SMBus and no RTC
CNN35XX-NFBE	CNN3510-NFBE-G (HW-1.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	No SMBus and no RTC

Model	Hardware Part Number (Version)	Firmware version	Distinguishing Features
CNN35XX-NFBE	CNL3560P-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3530-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510P-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560PB-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560B-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3530B-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510B-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510PB-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510LP-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510LPB-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC

Model	Hardware Part Number (Version)	Firmware version	Distinguishing Features
CNN35XX-NFBE	CNN3560P-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3560-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3530-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3505LP-NFBE-2.0-G (HW-2.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560P-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560B-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560B-NFBE-3.0-G-FB (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560A-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560C-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560D-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC

Model	Hardware Part Number (Version)	Firmware version	Distinguishing Features
CNN35XX-NFBE	CNL3560E-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3560F-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3530-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3530B-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3530A-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3530C-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3530D-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3530E-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3530F-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510P-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510A-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC

Model	Hardware Part Number (Version)	Firmware version	Distinguishing Features
CNN35XX-NFBE	CNL3510C-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510D-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510E-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510F-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNL3510I-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3560P-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3560-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3560A-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3560C-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3560D-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3560E-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3560F-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC

Model	Hardware Part Number (Version)	Firmware version	Distinguishing Features
CNN35XX-NFBE	CNN3530-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3530A-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3530C-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3530D-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3530E-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3530F-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510A-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510C-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510D-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510E-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510F-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC

Model	Hardware Part Number (Version)	Firmware version	Distinguishing Features
CNN35XX-NFBE	CNN3510LP-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510LPB-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510LPA-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510LPC-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510LPD-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510LPE-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3510LPF-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3505LP-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3505LPA-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3505LPC-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3505LPD-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC
CNN35XX-NFBE	CNN3505LPE-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC

Model	Hardware Part Number (Version)	Firmware version	Distinguishing Features
CNN35XX-NFBE	CNN3505LPF-NFBE-3.0-G (HW-3.0)	CNN35XX-NFBE-FW-2.09-0702 CNN35XX-UBOOT-4.03-03 CNN35XX-NFBE-SMW-2.09-0702	SMBus and RTC

2.2 Modes of Operation

The module supports the following modes of operation:

1. Non-Approved mode of operation
2. Approved Level 3 mode of operation

The module is initialized into one of the modes specified above during the module initialization period (see section 11 Life-Cycle Assurance for initialization procedure). The value of the parameter fipsState passed into the call specifies the mode. The following are the allowed values for fipsState parameters:

0 – Non-Approved mode

2 – Approved mode with single-factor authentication mechanism

3 – Approved mode with certificate based dual-factor authentication mechanism

The indicator of Approved mode is obtained by using the Get Status service. The fipsState field of Get Status service (CN_TOKEN_INFO) indicates the mode. CSPs are not shared between the Approved and non-Approved modes of operation.

2.3 Approved Mode of Operation

The module provides an Approved mode of operation, comprising all services described in Section 2.8 below. In this mode, the module allows only Approved or allowed algorithms. Request for any non-Approved/allowed algorithm is rejected.

2.4 Non-Approved Mode of Operation

The Module supports a Non-Approved mode implementing the non-Approved algorithms listed in Table 7. In this mode, the module also allows Approved or allowed algorithms.

2.5 Partitions

The module is an SR-IOV enabled intelligent PCIe adapter with 1 physical function and 128 virtual functions. In addition to the crypto offloads, this adapter can provide secure key storage with up to 64 partitions, including master partition. Each partition will have its own users to manage the partition and own configuration policies and hence each partition can be treated as a virtual HSM. HSM always has one default partition called HSM Master partition and this contains configuration of the complete HSM and default configuration of any additional partitions that are created. Only one HSM partition can be assigned to one SR-IOV virtual function of HSM adapter and vice-versa. Keys belonging to one partition are not accessible from other partition. This is achieved through a secure binding between partition and the PCIe virtual function.

2.5.1 HSM Master Partition

This is the default partition with only one user, called the Master Crypto Officer (MCO). This partition represents the operating state of the whole HSM adapter; i.e., initialization of HSM is nothing but initializing this partition with required configuration and MCO credentials. Zeroizing this partition will erase all HSM partitions in the adapter. The HSM must be initialized and the MCO should already be logged in to create more partitions on the adapter. The MCO can backup and restore complete partition including user data, partition configuration and user keys. All the backup data is encrypted with Backup keys.

2.5.2 HSM Partition

Each partition will have a different set of users to manage it and a dedicated key storage and crypto resources associated. A partition will have a default configuration supplied by the master partition and can be changed (within limits) during the partition initialization. When a partition is created by the MCO, it will be in a zeroized state and has to be initialized to do any keystore management or crypto function offloads. Partition initialization will create the Partition Crypto Officer (PCO). The PCO can later create up to 1024 users (PCO or PCU) on demand. Each user will have a unique user name to identify themselves. The User has to login to the partition/vHSM to issue any authorized commands. Users are authenticated using passwords submitted during the user creation.

2.6 Encrypted Communication Channels

The End-to-End encryption feature in the module allows an application to initiate an TLS connection with the firmware to ensure the confidentiality of the data communicated over PCIe path.

The connection is based on **TLS v1.2** with the cipher-suite **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256** and **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384** (known to OpenSSL as **AES128-GCM-SHA256, AES256-GCM-SHA384, ECDHE_RSA_AES128-GCM-SHA256**, and **ECDHE_RSA_AES256-GCM-SHA384**). The module will act as server, and host application will act as client. The **server private key** will be the partition private key PAK which is generated for each pHSM when the pHSM/partition is created. The **server certificate** used for the SSL connection is the partition certificate PAC. The complete chain will be validated by the host application (CavClient) before establishing the TLS connection.

The End-to-End encryption feature is enabled using the initialization configuration parameters. Once this feature is enabled, all commands except the initialize and open session are encrypted.

2.7 Supported Cryptographic Algorithms

This section provides the list of supported cryptographic algorithms segregated based on the operating mode.

2.8 Approved Algorithms

The cryptographic module supports the following Approved algorithms. Only the algorithms/modes listed in the table below are used in the module.

Note: All symmetric key sizes represent the key strength.

Table 4 – Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
A1190	**AES-CMAC (SP 800-38B)**	CMAC	AES CMAC for larger payloads Sizes: 128-bit, 192-bit and 256-bit Uses AES-CBC (#C819) as underlying Block cipher	Message authentication code generate/verify.
A1191	**KDF SP 800-108 (KBKDF)**	KDF SP 800-108	AES counter KDF 128-bit, 192-bit, and 256-bit Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator Uses AES-CMAC (#A1190) as the underlying PRF	Key derivation
A1192	**KDA HKDF Sp800-56Cr1**	KDA HKDF Sp800-56Cr1	KDA HKDF Shared Secret length: 224 – 4096, HMAC: SHA2-224, SHA2-256, SHA2-384, SHA2-512, Derived Key Length: 2048 Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator Uses HMAC-SHA2-224 (#C822), HMAC-SHA2-256 (#C822), HMAC-SHA2-384 (#C822) and HMAC-SHA2-512 (#C822) as underlying MAC algorithm	Key derivation and ECDH-AES key wrap
A1192	**KDA OneStep Sp800-56Cr1**	KDA OneStep Sp800-56Cr1	KDA OneStep Auxiliary functions: SHA2-224, SHA2-256, SHA2-384 and SHA2-512, Derived Key Length: 2048 Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator	Key derivation and ECDH-AES key wrap

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			Uses HMAC-SHA2-224 (#C822), HMAC-SHA2-256 (#C822), HMAC-SHA2-384 (#C822) and HMAC-SHA2-512 (#C822) as underlying MAC algorithm	
A1192	**KDA TwoStep Sp800-56Cr1**	KDA TwoStep Sp800-56Cr1	KDA TwoStep (HMAC and CMAC) MAC Salting Methods: random, Supported Lengths: 1-4096, KDF Mode: counter, MAC Modes: CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, Counter Lengths: 8, 16, 24, 32, Derived Key Length: 4096 Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator Uses HMAC-SHA2-224 (#C822), HMAC-SHA2-256 (#C822), HMAC-SHA2-384 (#C822) and HMAC-SHA2-512 (#C822) as underlying MAC algorithm Uses AES-CBC (#C819) as underlying Block cipher for AES-CMAC	Key derivation and ECDH-AES key wrap
A1193	**KAS-IFC-SSC (SP 800-56Br2)**	KAS-IFC-SSC	RSA 2048-bit, 3072-bit, 4096-bit 2048, 3072, and 4096-bit modulus providing 112, 128, and 150 bits of encryption strength respectively • RSA-based shared secret computation, KAS1 and KAS2 • Uses Counter DRBG (SP 800-90Ar1) (#C821) as	PEK and KLK generation and certificate authentication

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			underlying Random Generator • Uses HMAC-SHA2-224 (#C822), HMAC-SHA2-256 (#C822), HMAC-SHA2-384 (#C822) and HMAC-SHA2-512 (#C822) as underlying MAC algorithm • Uses RSA KeyGen (FIPS 186-4) [#C824] as underlying RSA key generation algorithm	
A1194	**KTS-IFC (KTS) (SP 800-56Br2)**	SP 800-56Brev2. KTS-IFC (key encapsulation and un-encapsulation) per IG D.G.	2048, 3072, or 4096-bit modulus providing 112, 128, or 150 bits of encryption strength respectively	Asymmetric key encapsulation and un-encapsulation in hybrid environment
A1194	**KTS-IFC (SP 800-56Br2)**	KTS-IFC	RSA key wrap and unwrap of symmetric keys in KTS-OAEP-Basic padding. 2048, 3072, 4096-bit modulus 2048, 3072, and 4096-bit modulus providing 112, 128, and 150 bits of encryption strength respectively • Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator • Uses HMAC-SHA2-224 (#C822), HMAC-SHA2-256 (#C822), HMAC-SHA2-384 (#C822) and HMAC-SHA2-512 (#C822) as underlying MAC algorithm • Uses RSA KeyGen (FIPS 186-4) [#C824] as the underlying Key generation algorithm	Asymmetric key encapsulation and un-encapsulation in hybrid environment
A1195	**AES-CMAC (SP 800-38B)**	AES-CMAC	Key Sizes 128-bit, 192-bit, and 256-bit	Used to compute key checksum value (KCV)

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			Uses AES-CBC (#C819) as underlying Block cipher for AES-CMAC	
A1196	**PBKDF (SP 800-132)**	PBKDF	HMAC with SHA-1, SHA2-224, SHA2-256, SHA2-384 and SHA2-512 Uses HMAC-SHA-1 (#C822 ,HMAC-SHA2-224 (#C822), HMAC-SHA2-256 (#C822), HMAC-SHA2-384 (#C822) and HMAC-SHA2-512 (#C822) as underlying MAC algorithm	User credentials storage
A1197	**SHA3-224 (FIPS 202)**	SHA3-224	SHA3-224	Message digests
A1197	**SHA3-256 (FIPS 202)**	SHA3-256	SHA3-256	Message digests
A1197	**SHA3-384 (FIPS 202)**	SHA3-384	SHA3-384	Message digests
A1197	**SHA3-512 (FIPS 202)**	SHA3-512	SHA3-512	Message digests
A1197	**SHAKE-128 (FIPS 202)**	SHAKE-128	SHAKE-128	Message digests
A1197	**SHAKE-256 (FIPS 202)**	SHAKE-256	SHAKE-256	Message digests
A1199	**RSA KeyGen (FIPS 186-4)**	RSA KeyGen (FIPS 186-4)	Key Generation Mode: B.3.3 Properties: Modulo: 4096 Primality Tests: C.2 Public Exponent Mode: Random Private Key Format: Standard Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator	Key generation
A1199	**RSA SigGen (FIPS 186-4)**	RSA SigGen (FIPS 186-4)	Signature Type: PKCS 1.5 Modulo: 4096 (SHA2-224 SHA2-256, SHA2-384, SHA2-512)	Sign

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			Signature Type: PKCSPSS Modulo: 4096 (SHA2-224 Salt Length: 28, SHA2-256 Salt Length: 32, SHA2-384 Salt Length: 48, SHA2-512 Salt Length: 64) Uses SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm 4096-bit modulus providing 150 bits of encryption strength	
A1199	**RSA SigVer (FIPS 186-4)**	RSA SigVer (FIPS 186-4)	Signature Type: PKCS 1.5 Modulo: 4096 (SHA2-224 SHA2-256, SHA2-384, SHA2-512) Signature Type: PKCSPSS Modulo: 4096 (SHA2-224 Salt Length: 28, SHA2-256 Salt Length: 32, SHA2-384 Salt Length: 48, SHA2-512 Salt Length: 64) Public Exponent Mode: Random Uses SHA-1 (#C820), SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm 4096-bit modulus providing 150 bits of encryption strength	Verify
A1200	**RSA Decryption Primitive (CVL) (SP 800-56Br2)**	RSA decryption primitive	2048-bit 3072 bit (CAVP testing was not available at the time of module submission) 4096-bit (CAVP testing was not available at the time of module submission) 2048, 3072, and 4096-bit modulus providing 112,	RSA key transport

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			128, and 150 bits of encryption strength respectively.	
A1201	**RSA SigVer (FIPS 186-4)**	RSA SigVer (FIPS 186-4)	RSA PKCS 1.5 2048-bit and SHA2-256 signature verification Uses SHA2-256 (#C820) as underlying digest algorithm 2048-bit modulus providing 112 bits of encryption strength	Firmware integrity verification by bootloader
A1202	**SHA2-256 (FIPS 180-4)**	SHA2-256	SHA2-256	Message Digest for Firmware image integrity verification by bootloader
A1203	**AES-GCM (KTS) (SP 800-38D)**	SP 800-38D and SP 800-38F. KTS (key wrapping and unwrapping) per IG D.G.	128, 192, and 256- bit keys providing 128, 192, or 256 bits of encryption strength	Data encryption, decryption, key-wrap, and key-unwrap
A1203	**AES-GCM (SP 800-38D)**	AES-GCM	Encrypt/Decrypt; 128, 192, and 256-bit Uses AES-ECB (#C839) as the underlying block cipher	Data encryption, decryption
A1219	**KAS-ECC (KAS)** **Sp800-56Ar3**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2).	P-521 curve providing 256 bits of encryption strength	Cloning
A1219	**KAS-ECC Sp800-56Ar3**	KAS-ECC Sp800-56Ar3	P-521, SHA2-512, and HMAC SHA2-512 Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator for the ECDSA KeyGen key pair P-521 curve providing 256 bits of encryption strength Uses ECDSA KeyVer (FIPS 186-4) (#C825) and ECDSA KeyGen (FIPS 186-4) (#C825) underlying verification for the key pair	Cloning

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			ECC KAS Ephemeral Unified with no Key confirmation using One-Step KDF Uses SHA2-512 (FIPS 180-4) (#C820) as the underlying digest algorithm	
A1220	**KAS-ECC-SSC Sp800-56Ar3**	KAS-ECC-SSC Sp800-56Ar3	ECDH: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 P-224, P-256, P384 and P-521 providing 112 bits,128 bits. 192 bits and 256 bits of encryption strength respectively Uses ECDSA KeyVer (FIPS 186-4) (#C825) and ECDSA KeyGen (FIPS 186-4) (#C825 underlying verification for the key pair Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator for the ECDSA KeyGen key pair Uses SHA2-256 (FIPS 180-4) (#C820), SHA2-384 (FIPS 180-4) (#C820), SHA2-512(FIPS 180-4) (#C820) as the underlying digest algorithm	Shared secret computation
A2161	**KAS-ECC-SSC Sp800-56Ar3**	KAS-ECC-SSC Sp800-56Ar3	ECDH: P-224, P-256, P-384, P-521 P-224, P-256, P384 and P-521 providing 112 bits,128 bits. 192 bits and 256 bits of encryption strength respectively Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator for the ECDSA KeyGen key pair	Shared secret computation

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			Uses ECDSA KeyVer (FIPS 186-4) (#C825) and ECDSA KeyGen (FIPS 186-4) (#C825 underlying verification for the key pair Uses SHA2-256 (FIPS 180-4) (#C820), SHA2-384 (FIPS 180-4) (#C820), SHA2-512(FIPS 180-4) (#C820) as the underlying digest algorithm	
C1169	**TDES-ECB (SP 800-38A)**	Triple-DES-ECB	3-key DES (192 bits) Decrypt • Key size 192 bits • Provides 112-bits of encryption strength * Legacy use only	Used as a pre-requisite for the TKW
C1263	**AES-KW (KTS) (SP 800-38F)**	SP 800-38F. KTS (key wrapping and unwrapping) per IG D.G.	128, 192, and 256- bit keys providing 128, 192, or 256 bits of encryption strength Uses AES-CBC (#C819) as underlying Block cipher	Key wrapping/unwrapping
C1263	**AES-KW (SP 800-38F)**	AES-KW	KW, 128, 192, and 256-bit Uses AES-CBC (#C819) as underlying Block cipher	Key wrapping/unwrapping
C1263	**AES-KWP (KTS) (SP 800-38F)**	SP 800-38F. KTS (key wrapping and unwrapping) per IG D.G.	128, 192, and 256- bit keys providing 128, 192, or 256 bits of encryption strength Uses AES-CBC (#C819) as underlying Block cipher	Key wrapping/unwrapping
C1263	**AES-KWP (SP 800-38F)**	AES-KWP	KWP, 128, 192, and 256-bit Uses AES-CBC (#C819) as underlying Block cipher	Key wrapping/unwrapping
C1263	**TDES-KW (SP 800-38F)**	Triple-DES-KW	TKW key size 192 bits Uses TDES-ECB (#C1169) as underlying Block cipher * Legacy use only	Key unwrapping
C1263	**Triple-DES-KW (KTS) (SP 800-38F)**	SP 800-38F. KTS (key unwrapping) per IG D.G.	192-bit keys providing 112 bits of encryption strength.	Key unwrapping

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			Uses TDES-ECB (#C1169) as underlying Block cipher * Legacy use only	
C819	**AES-CBC (SP 800-38A)**	AES-CBC	CBC mode: Encrypt, Decrypt; 128, 192, and 256-bit* *Encrypt only uses 256-bit	Encryption/Decryption
C819	**AES-ECB (SP 800-38A)**	AES-ECB	ECB mode: Encrypt, Decrypt; 128, 192, and 256-bit	Encryption/Decryption
C820	**SHA-1 (FIPS 180-4)**	SHA-1	SHA-1	Used in digests, HMAC, signature verification, and KDFs
C820	**SHA2-224 (FIPS 180-4)**	SHA2-224	SHA2-224	Used in digests, HMAC, signature generation/ verification, and KDFs
C820	**SHA2-256 (FIPS 180-4)**	SHA2-256	SHA2-256	Used in digests, HMAC, signature generation/ verification, and KDFs
C820	**SHA2-384 (FIPS 180-4)**	SHA2-384	SHA2-384	Used in digests, HMAC, signature generation/ verification, and KDFs
C820	**SHA2-512 (FIPS 180-4)**	SHA2-512	SHA2-512	Used in digests, HMAC, signature generation/ verification, and KDFs
C821	**Counter DRBG (SP 800-90Ar1)**	Counter DRBG	AES 256 with df • No prediction resistance • Uses AES-CBC (#C819) as underlying Block cipher	Random number generation for user, internal IVs and salt
C822	**HMAC-SHA-1 (FIPS 198-1)**	HMAC-SHA-1	HMAC-SHA-1 Uses SHA-1 (#C820) as underlying digest algorithm Keys size of 160 bits	MAC generation, verify, KAS and KDF
C822	**HMAC-SHA2-224 (FIPS 198-1)**	HMAC-SHA2-224	HMAC-SHA2-224 Uses SHA2-224 (#C820) as underlying digest algorithm Key size of 224 bits	MAC generation, verify, KAS and KDF

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
C822	**HMAC-SHA2-256 (FIPS 198-1)**	HMAC-SHA2-256	HMAC-SHA2-256 Uses SHA2-256 (#C820) as underlying digest algorithm Key size of 256 bits	MAC generation, verify, KAS and KDF
C822	**HMAC-SHA2-384 (FIPS 198-1)**	HMAC-SHA2-384	HMAC-SHA2-384 Uses SHA2-384 (#C820) as underlying digest algorithm Key size of 384 bits	MAC generation, verify, KAS and KDF
C822	**HMAC-SHA2-512 (FIPS 198-1)**	HMAC-SHA2-512	HMAC-SHA2-512 Uses SHA2-512 (#C820) as underlying digest algorithm Key size of 512 bits	MAC generation, verify, KAS and KDF
C823	**DSA KeyGen (FIPS 186-4)**	DSA KeyGen (FIPS 186-4)	Key Gen: 2048 and 3072-bit Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator for the key generation	Key generation
C823	**DSA PQGGen (FIPS 186-4)**	DSA PQGGen (FIPS 186-4)	PQG Gen: 2048 and 3072-bit Uses SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm Provides the encryption strength of 112 bits and 128 bits for the 2048 bit and 3072 bits key size Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator for the key generation	Domain parameter generation
C823	**DSA PQGVer (FIPS 186-4)**	DSA PQGVer (FIPS 186-4)	PQG Ver: 1024-bit*, 2048 and 3072-bit	Domain parameter verification

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			Uses SHA-1 (#C820), SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm Provides the encryption strength of 112 bits and 128 bits for the 2048 bit and 3072 bits key size * Legacy use only	
C823	**DSA SigGen (FIPS 186-4)**	DSA SigGen (FIPS 186-4)	Sig Gen: 2048 and 3072-bit (SHA2-224, 256, -384, -512) Uses SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm Provides the encryption strength of 112 bits and 128 bits for the 2048 bit and 3072 bits key size Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator for the key generation	Sign
C823	**DSA SigVer (FIPS 186-4)**	DSA SigVer (FIPS 186-4)	Sig Ver: 1024*, 2048 and 3072-bit (SHA-1, 224, -256, -384, -512) Uses SHA-1 (#C820), SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm Provides the encryption strength of 80 bits, 112 bits and 128 bits for the 1024 bits, 2048 bit and 3072 bits key size * Legacy use only	Verify
C824	**RSA KeyGen (FIPS 186-4)**	RSA KeyGen (FIPS 186-4)	KeyGen: 2048, 3072-bit	Key generation

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator 2048 and 3072 modulus providing 112 and 128 bits of encryption strength	
C824	**RSA SigGen (FIPS 186-4)**	RSA SigGen (FIPS 186-4)	FIPS 186-4 PKCS #1 1.5 and PSS SigGen: 2048 and 3072-bit (SHA2-224, -256, -384, -512) Uses SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm 2048 and 3072 modulus providing 112 and 128 bits of encryption strength	Sign
C824	**RSA SigVer (FIPS 186-4)**	RSA SigVer (FIPS 186-4)	FIPS 186-4 PKCS #1 1.5 and PSS SigVer: 1024, 2048 and 3072-bit (SHA-1, 224, -256, -384, -512) Uses SHA-1 (#C820), SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm 2048 and 3072 modulus providing 112 and 128 bits of encryption strength	Verify
C825	**ECDSA KeyGen (FIPS 186-4)**	ECDSA KeyGen (FIPS 186-4)	Key Gen: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator	Key generation
C825	**ECDSA KeyVer (FIPS 186-4)**	ECDSA KeyVer (FIPS 186-4)	Key Ver; All P, K and B curves	Key verification

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
C825	**ECDSA SigGen (FIPS 186-4)**	ECDSA SigGen (FIPS 186-4)	Sig Gen: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 (SHA2-224, -256, -384, -512) Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator Uses SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm P-224, P-256, P-384, and P-521 curves providing 112, 128, 192, or 256 bits of encryption strength respectively	Signature generation
C825	**ECDSA SigGen (FIPS 186-4) (CVL)**	ECDSA SigGen (FIPS 186-4)	Sig Gen Component: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 (SHA2-224, -256, -384, -512) Uses Counter DRBG (SP 800-90Ar1) (#C821) as underlying Random Generator P-224, P-256, P-384, and P-521 curves providing 112, 128, 192, or 256 bits of encryption strength respectively	Signature generation
C825	**ECDSA SigVer (FIPS 186-4)**	ECDSA SigVer (FIPS 186-4)	SigVer: All P, K and B curves (SHA-1, 224, 256, -384, -512) Uses SHA-1 (#C820), SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm	Signature verification

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			P-192 ,P-224, P-256, P-384, and P-521 curves providing 80, 112, 128, 192, or 256 bits of encryption strength respectively	
C825	**KDF ANS 9.63 (CVL) (SP 800-135r1)**	KDF ANS 9.63	(SHA2-224, SHA2-256, SHA2-384, SHA2-512) Uses SHA2-224 (#C820), SHA2-256 (#C820), SHA2-384 (#C820) and SHA2-512 (#C820) as underlying digest algorithm	Key derivation and key agreement schemes
C826	**KDF SP 800-108 (KBKDF)**	KDF SP 800-108	HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA 2-256, HMAC-SHA 2-384, HMAC-SHA 2-512 KDF Uses HMAC-SHA-1 (#C822), HMAC-SHA2-224 (#C822), HMAC-SHA2-256 (#C822), HMAC-SHA2-384 (#C822) and HMAC-SHA2-512 (#C822) as underlying MAC algorithm	Key derivation
C829	**ECDSA SigGen (FIPS 186-4) (CVL)**	ECDSA SigGen (FIPS 186-4)	Sig Gen Component: P-224, P-256, P-384, P-521 (SHA2-224, -256, -384, -512) Uses Hash DRBG (SP 800-90Ar1) (#C830) as underlying Random Generator P-224, P-256, P-384, and P-521 curves providing 112, 128, 192, or 256 bits of encryption strength respectively	Signature generation
C829	**ECDSA SigVer (FIPS 186-4)**	ECDSA SigVer (FIPS 186-4)	SigVer: P-192, P-224, P-256, P-384, P-521 (SHA-1, 224, -256, 384, -512)	Signature verification

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			Uses SHA-1 (# SHS 1780), SHA2-224 (# SHS 1780), SHA2-256 (# SHS 1780), SHA2-384 (# SHS 1780) and SHA2-512 (# SHS 1780) as underlying digest algorithm P-192, P-224, P-256, P-384, and P-521 curves providing 80, 112, 128, 192, or 256 bits of encryption strength respectively	
C829	**KAS-ECC CDH-Component (CVL) (SP 800-56Ar3)**	KAS-ECC CDH-Component	P-224, P-256, P-384 and P-521 P-224, P-256, P-384, and P-521 curves providing 112, 128, 192, or 256 bits of encryption strength respectively	ECDH key derivation and SSL suite B key exchange
C830	**Hash DRBG (SP 800-90Ar1)**	Hash DRBG	SHA2-512 based with security strength of 256-bit. No prediction resistance Uses SHA2-512 (# SHS 1780) as underlying digest algorithm	Random number generation for user, internal Ivs and salt
C839	**AES-CBC (SP 800-38A)**	AES-CBC	Encrypt/Decrypt; 128, 192 and 256-bit	Data encryption and decryption
C839	**AES-CCM (SP 800-38C)**	AES-CCM	Authenticated encryption and decryption; 128-bit, 192-bit, and 256-bit Uses AES-CBC (#C839) as underlying Block cipher	Data encryption and decryption
C839	**AES-CMAC (SP 800-38B)**	AES-CMAC	MAC generate and verify; 128-bit, 192-bit, and 256-bit Uses AES-CBC (#C839) as underlying Block cipher	Message authentication code generation and verification
C839	**AES-CTR (SP 800-38A)**	AES-CTR	Encrypt/Decrypt 128, 192, and 256-bit Uses AES-ECB (#C839) as underlying Block cipher	Data encryption and decryption

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
C839	**AES-ECB (SP 800-38A)**	AES-ECB	Encrypt/Decrypt; 128, 192, and 256-bit	Data encryption and decryption
C839	**AES-GCM (KTS) (SP 800-38D)**	SP 800-38D and SP 800-38F. KTS (key wrapping and unwrapping) per IG D.G.	128, 192, and 256- bit keys providing 128, 192, or 256 bits of encryption strength Uses AES-ECB (#C839) as underlying Block cipher	Data encryption, decryption, key-wrap, and key-unwrap
C839	**AES-GCM (SP 800-38D)**	AES-GCM	Encrypt/Decrypt; 128, 192, and 256-bit Uses AES-ECB (#C839) as underlying Block cipher	Data encryption, decryption, key-wrap, and key-unwrap
C839	**AES-GMAC (SP 800-38D)**	AES-GMAC	Encrypt/Decrypt; 128, 192, and 256-bit Uses AES-ECB (#C839) as underlying Block cipher	Message Authentication
C839	**HMAC-SHA-1 (FIPS 198-1)**	HMAC-SHA-1	HMAC-SHA-1 Uses SHA-1 (# SHS 1780) as underlying digest algorithm	MAC generation, verify, KAS and KDF
C839	**HMAC-SHA2-224 (FIPS 198-1)**	HMAC-SHA2-224	HMAC-SHA2-224 Uses SHA2-224 (# SHS 1780) as underlying digest algorithm	MAC generation, verify, KAS and KDF
C839	**HMAC-SHA2-256 (FIPS 198-1)**	HMAC-SHA2-256	HMAC-SHA2-256 Uses SHA2-256 (# SHS 1780) as underlying digest algorithm	MAC generation, verify, KAS and KDF
C839	**HMAC-SHA2-384 (FIPS 198-1)**	HMAC-SHA2-384	HMAC-SHA2-384 Uses SHA2-384 (# SHS 1780) as underlying digest algorithm	MAC generation, verify, KAS and KDF
C839	**HMAC-SHA2-512 (FIPS 198-1)**	HMAC-SHA2-512	HMAC-SHA2-512 Uses SHA2-512 (# SHS 1780) as underlying digest algorithm	MAC generation, verify, KAS and KDF
C839	**KDF SP 800-108 (KBKDF)**	KDF SP 800-108	HMAC-Counter mode (HMAC-SHA-256, HMAC-SHA2-384, HMAC-SHA2-512)	Key derivation

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
			CMAC counter mode (128-bit, 192-bit, and 256-bit)	
C839	**RSA Decryption Primitive (CVL) (SP 800-56Br2)**	RSA Decryption Primitive	2048-bit 3072-bit 4096-bit **Note**: CAVP testing was only available for 2048-bit at the time of the module submission SP 800-56B RSADP component validation is equivalent with SP 800-56Br2 2048, 3072, and 4096-bit modulus providing 112, 128, and 150 bits of encryption strength	Decryption primitive
C839	**RSA Signature Primitive (CVL) (FIPS 186-4)**	RSA Signature Primitive	2048-bit 3072-bit 4096-bit **Note**: CAVP testing was only available for 2048-bit at the time of the module submission 2048, 3072, and 4096-bit modulus providing 112, 128, and 150 bits of encryption strength.	Signature primitive
C840	**KDF TLS (CVL) (SP 800-135r1)**	KDF TLS	TLS-KDF (v1.0/1.1, v1.2) v1.2: SHA2-256, SHA2-384, SHA2-512	TLS handshake
KAS-ECC-SSC Sp800-56Ar3 (#A2161) KDA HKDF SP 800-56Cr1 (#A1192)	**KAS KDA HKDF (SP 800-56Ar3)**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2).	P-224, P-256, P-384, and P-521 curves providing 112, 128, 192, or 256 bits of encryption strength	ECDH key derivation and ECDH-AES key wrap

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
KAS-ECC-SSC Sp800-56Ar3 (#A2161) KDA OneStep SP 800-56Cr1 (#A1192)	**KAS KDA ONESTEP (SP 800-56Ar3)**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2).	P-224, P-256, P-384, and P-521 curves providing 112, 128, 192, or 256 bits of encryption strength	ECDH key derivation and ECDH-AES key wrap
KAS-ECC-SSC Sp800-56Ar3 (#A2161) KDA TwoStep SP 800-56Cr1 (#A1192)	**KAS KDA TWOSTEP (SP 800-56Ar3)**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2).	P-224, P-256, P-384, and P-521 curves providing 112, 128, 192, or 256 bits of encryption strength	ECDH key derivation and ECDH-AES key wrap
KAS-ECC-SSC Sp800-56Ar3 (#A1220) KDA HKDF Sp800-56Cr1 (#A1192)	**KAS-KDF-HKDF (SP 800-56Ar3)**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2)	P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 curves providing 112, 128, 192, or 256 bits of encryption strength	ECDH key derivation and ECDH-AES key wrap
KAS-ECC-SSC Sp800-56Ar3 (#A1220) KDA OneStep Sp800-56Cr1 (#A1192)	**KAS-KDF-OneStep(SP 800-56Ar3)**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2)	P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 curves providing 112, 128, 192, or 256 bits of encryption strength	ECDH key derivation and ECDH-AES key wrap
KAS-ECC-SSC Sp800-56Ar3 (#A1220) KDA TwoStep Sp800-56Cr1 (#A1192)	**KAS-KDF-TwoStep (SP 800-56Ar3)**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2)	P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 curves providing 112, 128, 192, or 256 bits of encryption strength	ECDH key derivation and ECDH-AES key wrap

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
KAS-ECC-SSC Sp800-56Ar3 (#A1220) KDF ANS 9.63 (#C825)	**KAS-KDF-ANS9.63 (SP 800-56Ar3)**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2)	P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 curves providing 112, 128, 192, or 256 bits of encryption strength	ECDH key derivation and ECDH-AES key wrap
KAS-ECC-SSC Sp800-56Ar3 (#A2161) KDF ANS 9.63 (CVL) (SP 800-135r1) (#C825)	**KAS ANS 9.63 (SP 800-56Ar3)**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2).	P-224, P-256, P-384, and P-521 curves providing 112, 128, 192, or 256 bits of encryption strength	ECDH key derivation and ECDH-AES key wrap
KAS-ECC-SSC Sp800-56Ar3 (#A2161) KDF TLS (CVL) (#C840)	**KAS TLS (SP 800-56Ar3)**	SP 800-56Arev3. KAS-ECC per IG D.F Scenario 2 path (2)	P-224, P-256, P-384, P-521 curves providing 112, 128, 192, or 256 bits of encryption strength	TLS
KAS-IFC-SSC (#A1193) KDA HKDF Sp800-56Cr1 (#A1192)	**KAS-IFC HKDF (SP 800-56Br2)**	SP 800-56Brev2. KAS-IFC per IG D.F Scenario 1 path (2).	2048-bit, 3072-bit and 4096-bit modulus providing 112, 128, or 150 bits of encryption strength	PEK and KLK generation and certificate authentication
KAS-IFC-SSC (#A1193) KDA OneStep Sp800-56Cr1 (#A1192)	**KAS-IFC OneStep (SP 800-56Br2)**	SP 800-56Brev2. KAS-IFC per IG D.F Scenario 1 path (2).	2048-bit, 3072-bit and 4096-bit modulus providing 112, 128, or 150 bits of encryption strength	PEK and KLK generation and certificate authentication
KAS-IFC-SSC (#A1193) KDA TwoStep Sp800-56Cr1 (#A1193)	**KAS-IFC TwoStep (SP 800-56Br2)**	SP 800-56Brev2. KAS-IFC per IG D.F Scenario 1 path (2).	2048-bit, 3072-bit and 4096-bit modulus providing 112, 128, or 150 bits of encryption strength	PEK and KLK generation and certificate authentication

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s) / Key Strength(s)	Use/Function
N/A	**ENT (P) SP 800-90B**	N/A	OCTEON HW RBG	System Entropy
SHS 1780	**SHA-1 (FIPS 180-4)**	SHA-1	SHA-1	Message digests
SHS 1780	**SHA2-224 (FIPS 180-4)**	SHA2-224	SHA2-224	Message digests
SHS 1780	**SHA2-256 (FIPS 180-4)**	SHA2-256	SHA2-256	Message digests
SHS 1780	**SHA2-384 (FIPS 180-4)**	SHA2-384	SHA2-384	Message digests
SHS 1780	**SHA2-512 (FIPS 180-4)**	SHA2-512	SHA2-512	Message digests
TDES 1311	**TDES-CBC (SP 800-38A)**	Triple-DES-CBC	TCBC mode; 3-key (192 bits) Decrypt * Legacy use only	Data decryption
TDES 1311	**TDES-ECB (SP 800-38A)**	Triple-DES-ECB	TECB mode; 3-key (192 bits) Decrypt * Legacy use only	Data decryption
Vendor affirmed	**CKG SP 800-133Rev2**	Please refer to section 2.9 Algorithm-Specific Information CKG	Please refer to section 2.9 Algorithm-Specific Information	Cryptographic Key Generation; SP 800-133Rev2 and IG D.H

2.9 Algorithm-Specific Information

- AES-GCM (#A1203)
  - IG C.H Notes:
    - Ivs are generated randomly, and IG C.H Option #2 applies.
    - IV is generated internally to the cryptographic module.
    - SP 800-38D §8.2.2 is used for GCM IV construction.
    - IV's random field is a 128-bit random number.
    - For IV restoration conditions guidance, refer to section 11.4 User Guidance.
    - Approved RBG (Hash DRBG #C830): SP 800-90Ar1 DRBG, HASH_DRBG SHA2-512
- AES-GCM (#C839)
  - IG C.H Scenario #1:
    - TLS 1.2 or other applications can offload GCM operations.
    - For TLS-1.2 protocol, IV constructed as described in RFC 5288.
    - Refer Section 2.12 for the TLS 1.2 AES GCM supported cipher suites
    - IV is generated internally to the cryptographic module.
    - The module triggers a handshake to establish new encryption keys and Ivs when the IV exhausts the maximum possible values for the given session key.
    - SP 800-38D §8.2.2 is used for GCM IV construction.
  - IG C.H Scenario #2:
    - Ivs are generated randomly and IG C.H Option #2 applies.

        - IV’s free field is a 4-byte counter
        - IV’s random field is a 96-bit random number.
        - IV’s random field is incremented by 1. IV’s random field wouldn’t overflow 96-bits in the lifetime of the module.
        - For IV restoration conditions guidance, refer to section 11.4 User Guidance.
        - Internal Approved RBG (Hash DRBG #C830): SP 800-90A DRBG, HASH_DRBG SHA2-512.
- CKG SP 800-133Rev2 (Vendor affirmed)
    - IG D.H
        - SP 800-133Rev2 Section 5.1 Asymmetric signature key generation using unmodified DRBG output SP 800-133Rev2
        - SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output
        - SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output
        - SP 800-133Rev2 Section 6.2.1 Derivation of symmetric keys from a key agreement shared secret.
        - SP 800-133Rev2 Section 6.2.2 Derivation of symmetric keys from a pre-shared key
- PBKDF (SP 800-132) (#A1196)
    - PBKDF with HMAC password strength
        - The password is a minimum of 8 characters, case-sensitive alpha-numeric. As such there are (26*2+10)^8 = 62^8 possible minimum-length passwords, and the false acceptance rate is 1 in 62^8 which is less than 1 in 1,000,000.
        - A maximum of 20 password attempts are possible before permanent lockout. Therefore the probability of false authentication over any timeframe is 20 in 62^8, which is less than 1 in 100,000. (The number of allowed login attempts prior to lockout is configured during module initialization but cannot exceed 20.)
        - Lockout of MCO automatically zeroizes the module in the next reboot. In all other cases, lockout can be unset by deleting the partition.
    - PBKDF with HMAC Iteration Count and Justification
        - Iteration count should be at least 1000, following the recommendation in SP 800-132r2.
        - Salt length is 16 bytes

The cryptographic module supports the following non-Approved algorithms that are allowed for use in Approved mode.

Table 5 – Non-Approved Algorithms Allowed in the Approved Mode of Operation

Algorithm	Caveat	Use/Function
AES	Cert. #C819, key unwrapping. Provides 128, 192 or 256 bits of encryption strength. Per IG D.G.	Key unwrap only N3FIPS-OpenSSL-1.1.1-AES ECB mode: Decrypt; 128, 192 and 256 bits CBC mode: Decrypt: 128, 192 and 256 bits *Legacy use only

Algorithm	Caveat	Use/Function
EC Diffie-Hellman with non-NIST recommended curves	Cert. #C829, provides 112, 128, 160, 192 or 256 bits of encryption strength. Per IG C.A.	EC-DH Secp224k1(112 bits), Secp256K1 (128 bits) • Prime order curve, generated as per FIPS 186-4 Section 6.1.1 brainpoolP224r1(112 bits), brainpoolP256r1(128 bits), brainpoolP320r1(160 bits), brainpoolP384r1(192 bits), brainpoolP512r1(256 bits) FRP256v1 (128 bits) • Prime order curve, generated as per FIPS 186-4 Section 6.1.1 (SHA-1*, SHA2-224, SHA2-256, SHA2-384, SHA2-512)
ECDSA with non-NIST recommended curves	Cert. #C825, provides 112, 128, 160, 192 or 256 bits of encryption strength. Per IG C.A.	EC Key generation, sign, verify Secp224k1(112 bits), Secp256K1 (128 bits) • Prime order curve, generated as per FIPS 186-4 Section 6.1.1 brainpoolP224r1(112 bits), brainpoolP256r1(128 bits), brainpoolP320r1(160 bits), brainpoolP384r1(192 bits), brainpoolP512r1(256 bits) FRP256v1 (128 bits) • Prime order curve, generated as per FIPS 186-4 Section 6.1.1 (SHA-1*, SHA2-224, SHA2-256, SHA2-384, SHA2-512)

The support of TLS 1.0/1.1, v1.2 protocol by the module is restricted to the TLS Key Derivation Function and the crypto operation. This functionality of the module is used by the user of the module as part of TLS protocol negotiation. No parts of the TLS protocol, other than the KDF, have been reviewed or tested by the CAVP or the CMVP.

2.10 Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed

Table 6 – Non-Approved Algorithms Allowed in Approved Mode with No Security Claimed

Algorithm/Function	Caveat	Use/Function
MD5	Only allowed as the PRF in TLSv1.0 and v1.1 per IG 2.4.A	Message digest used in TLSv1.0 / v1.1 KDF only.
Triple-DES SP 800-38B	No security claimed per IG 2.4.A	TDES-CMAC Cert# A1198

Algorithm/Function	Caveat	Use/Function
		Key Sizes • 192-bit (Generation, Verify) • Used to compute key checksum value (KCV) and KCV serves as Fingerprint of the Key.

2.11 Non-Approved, Non-Allowed Algorithms

The cryptographic module supports the following non-Approved algorithms available only in non-Approved mode.

Table 7 – Non-Approved Algorithms Not Allowed in the Approved Mode of Operation

Algorithm/Function	Usage
AES (non-compliant)	• Key wrap (TR31/TR34/AES-CBC/AES-GCM wrap/unwrap), DecimalTable/Data/PIN encryption/decryption. • FF1/FF3-1 Data encryption/decryption • In Non-Approved mode AES GCM supports the IV length from 1 byte to 16 bytes
DES	• Derive unique key per transaction (DUKPT), • EMV key derivation • Derive PIN from Offset • Derive Offset from PIN • PIN Verification • PVV generation and Verification • CVV generation and verification • Export Symmetric key/Export Asymmetric key pair using TR31 wrapping. • Import/Export using TR34. • Import Decimal Table • EMV script. • EMV ARQC/ARPC • Data/PIN encryption/decryption
DES MAC	MAC generation and Verification

Algorithm/Function	Usage
Double-DES	• Derive unique key per transaction (DUKPT) • EMV key derivation • Derive PIN from Offset • Derive Offset from PIN • PIN Verification • PVV generation and verification • CVV generation and verification • Export Symmetric key/Export Asymmetric key pair using TR31 wrapping • Import/Export using TR34 • Import Decimal Table • EMV script. • EMV ARQC/ARPC • Data/PIN encryption/decryption
EC-AES	EC-AES wrap/unwrap (EC BYOK)
ECDH KDF	Key derivation using ECDH followed by HMAC/CMAC counter KDF
ECDSA (non-compliant)	Key generation, Sign, Verify P192, Secp192k1, brainpoolP160r1, brainpool192r1, K-163 and B-163 (SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512)
EDDSA (non-compliant)	Key generation, Sign, Verify
KAS-ECC (non-compliant)	EC Key generation and ECDH Curve25519 (128 bits), Curve448 (224 bits)
PBE	Key generation
RSA (non-compliant)	• TR34 Import • TR34 Export • PIN block decryption • BYOK • Encrypt/Decrypt • Asymmetric key encapsulation and un-encapsulation using PKCS#1-v1.5 padding with modulus size 2048, 3072, and 4096 bits • Key generation, Sign, Verify (1024-bit)
Shamir’s Key Share	• Key share

Algorithm/Function	Usage
Triple-DES (non-compliant)	• Derive unique key per transaction (DUKPT) • EMV key derivation • Derive PIN from Offset • Derive Offset from PIN • PIN Verification • PVV generation and Verification • CVV generation and verification • Export Symmetric key/Export Asymmetric key pair using TR31 wrapping • Import/Export using TR34 • Import Decimal Table • EMV script • EMV ARQC/ARPC • Data/PIN encryption/decryption

2.12 TLS 1.0/1.1/1.2 Cipher Suites

The module supports the algorithms for the following cipher suites using Approved and allowed algorithms and key sizes:

- TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

For cipher suites using GCM, the IV is generated per RFC 5288. The module supports GCM cipher suites compatible with SP 800-52 Rev2.

2.13 Module Photographs

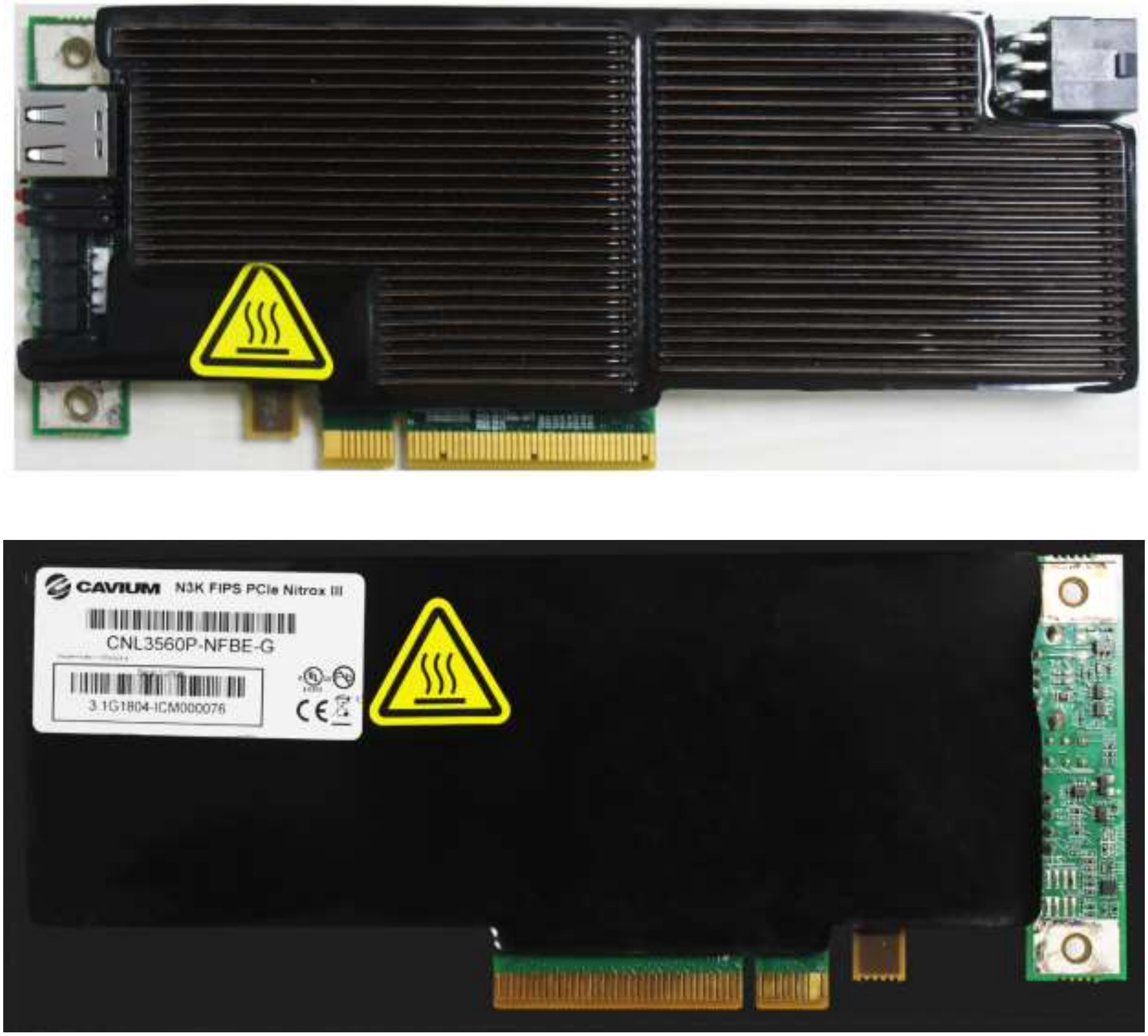


Figure 1 – Top View and Bottom View of Cryptographic Module (HW-1.0)

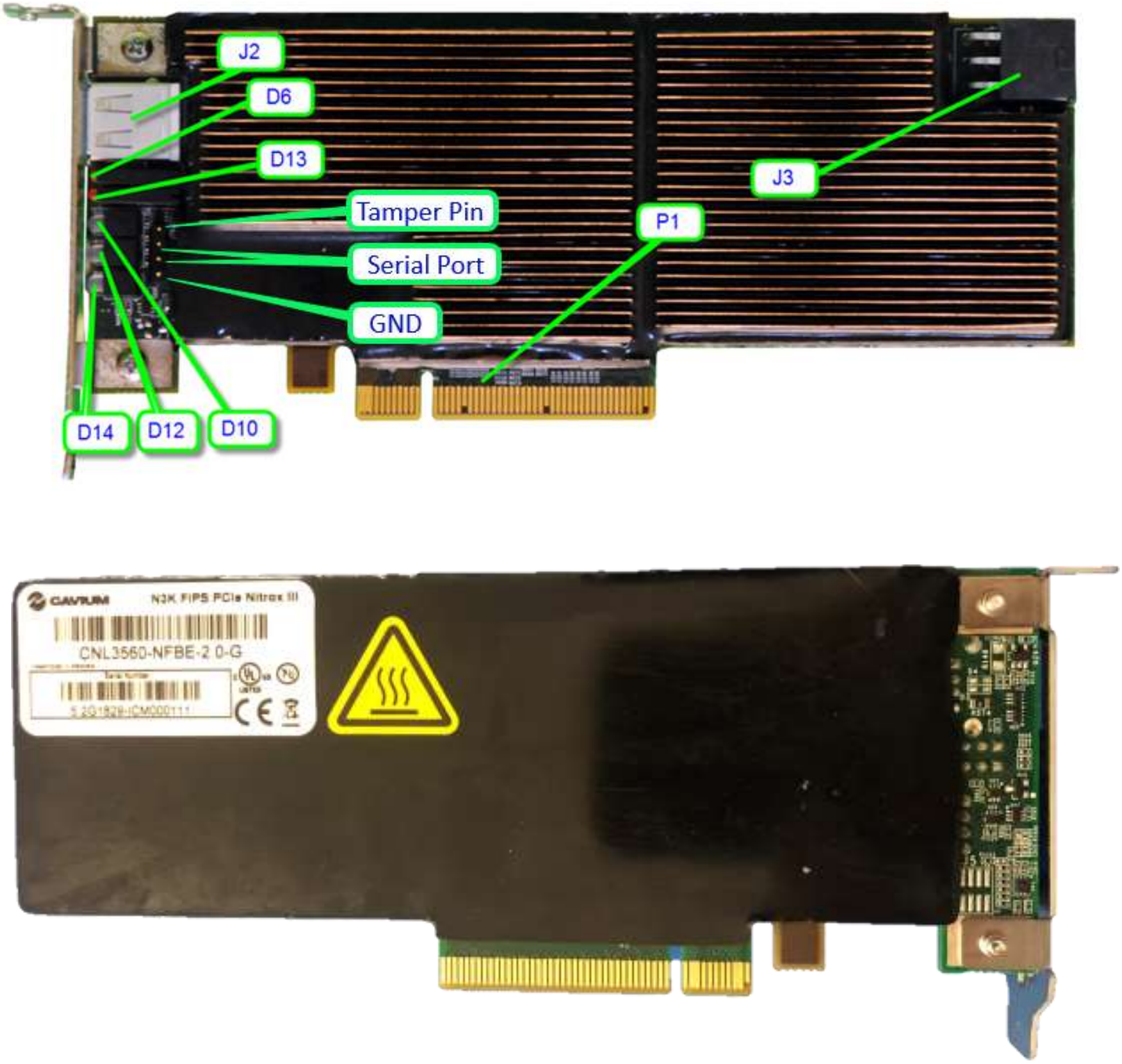


Figure 2 – Top View and Bottom View of Cryptographic Module (HW-2.0)

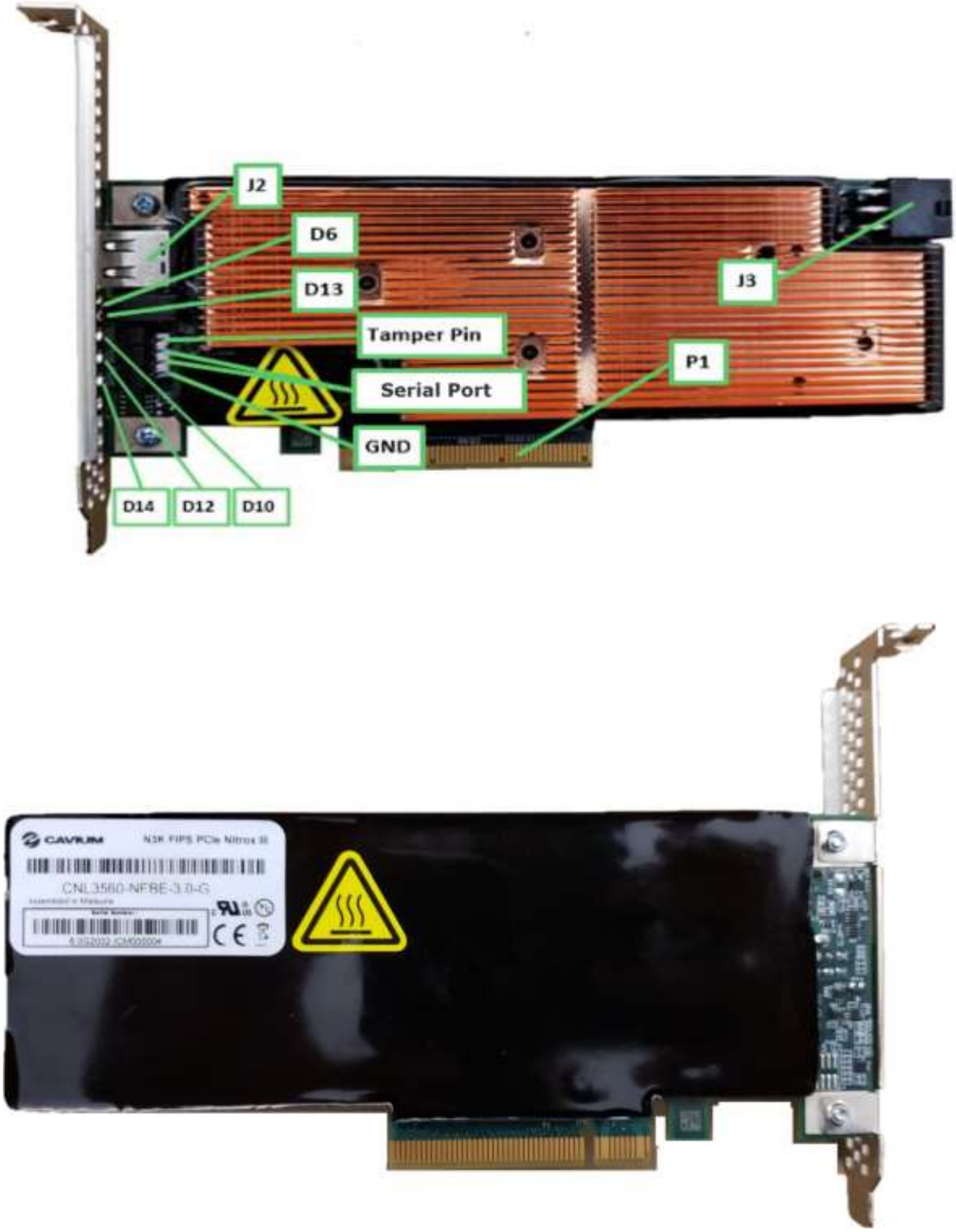


Figure 3 – Top View and Bottom View of Cryptographic Module (HW-3.0)

3 Cryptographic Module Interfaces

Table 8 describes the module ports and interfaces:

Table 8 – Ports and Interfaces

Physical port	Logical interface	Data that passes over port/ interface
PCIe Interface (P1)	Data Input and Data Output, Control Input, Status Output and Power. PCIE x8 Interface Lane 0 Transmit Side B (14, 15) Receive Side A (16, 17) Lane 1 Transmit Side B (19, 20) Receive Side A (21, 22) Lane 2 Transmit Side B (23, 24) Receive Side A (25, 26) Lane 3 Transmit Side B (27, 28) Receive Side A (29, 30) Lane 4 Transmit Side B (33, 34) Receive Side A (35, 36) Lane 5 Transmit Side B (37, 38) Receive Side A (39, 40) Lane 6 Transmit Side B (41, 42) Receive Side A (43, 44) Lane 7 Transmit Side B (45, 46) Receive Side A (47, 48)	Primary interface to communicate with the module Provides Services for the software on the host to communicate with the module
LED indicators	Status Output: LED interface (7 LEDs, 13 pins)	Visual status indicator
Tamper PIN	Control Input: Tamper pin GPIO over I2C	No data, only a signal from high to low
Power connector	Power: 6 PIN power connector	External power connector
USB Port (J2)	Data Input	Interface for vendor zeroize and Firmware Update (non-approved) service
Serial Port	Control Input and Status Output	Interface for vendor zeroize

Table 9 – Port/LED Description

LED/Port Location	Description
D6 – Red	Power Fail indication
D6 – Green	Power OK – All voltages rails are at nominal
D13 – Red	See Table 20
D13 – Green	See Table 20
D10 –Multicolor	See Table 20
D12 –Multicolor	See Table 20
D14 –Multicolor	See Table 20
GND	Ground PIN
P1	PCIE Interface see Table 8
J2	USB Port. To be used for vendor zeroize as an alternative to Tamper Pin zeroization (Under Vendor supervision).
J3	6 PIN Power Connector
Serial Port	To be used for vendor zeroize as an alternative to Tamper Pin zeroization (Under Vendor supervision). This will become read-only beyond bootloader.

4 Roles, Services, and Authentication

Note: Service interface documentation details specific service inputs and outputs:

Document name: LiquidSecurity-2.09-0702-Driver-APIs-html.zip
Version: 2.09-0702
Release date: 04/26/2024

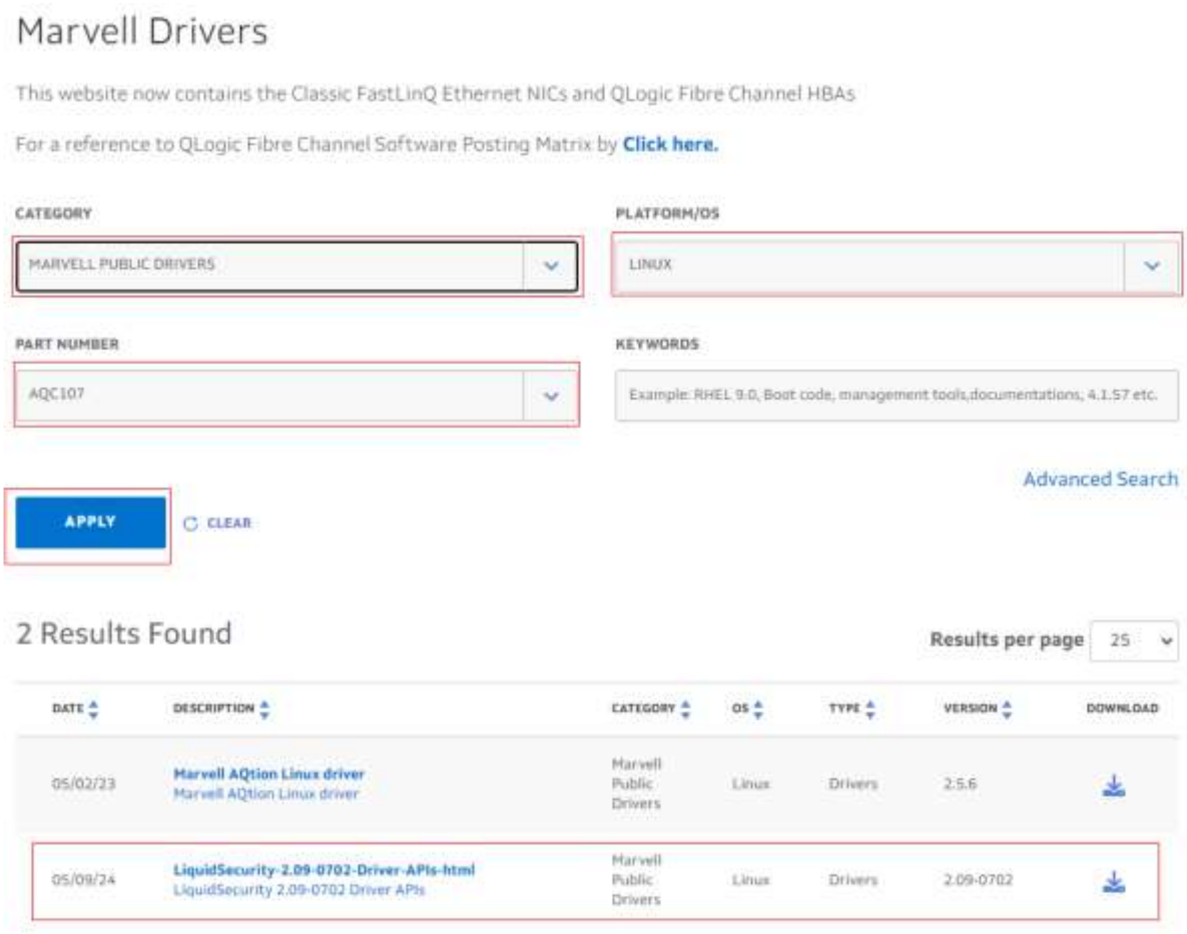
To access the document, complete the below steps.

1. Open the following link to open the Marvell Public Driver Downloads page:

   https://www.marvell.com/support/downloads.html#

2. Choose CATEGORY, PLATFORM, and PART NUMBER as shown in the following screenshot; then click the **APPLY** button.

   Two results will display; select “LiquidSecurity-2.09-0702-Driver-APIs-html" to download.



Marvell Drivers

This website now contains the Classic FastLinQ Ethernet NICs and QLogic Fibre Channel HBAs

For a reference to QLogic Fibre Channel Software Posting Matrix by **Click here.**

CATEGORY: MARVELL PUBLIC DRIVERS

PLATFORM/OS: LINUX

PART NUMBER: AQC107

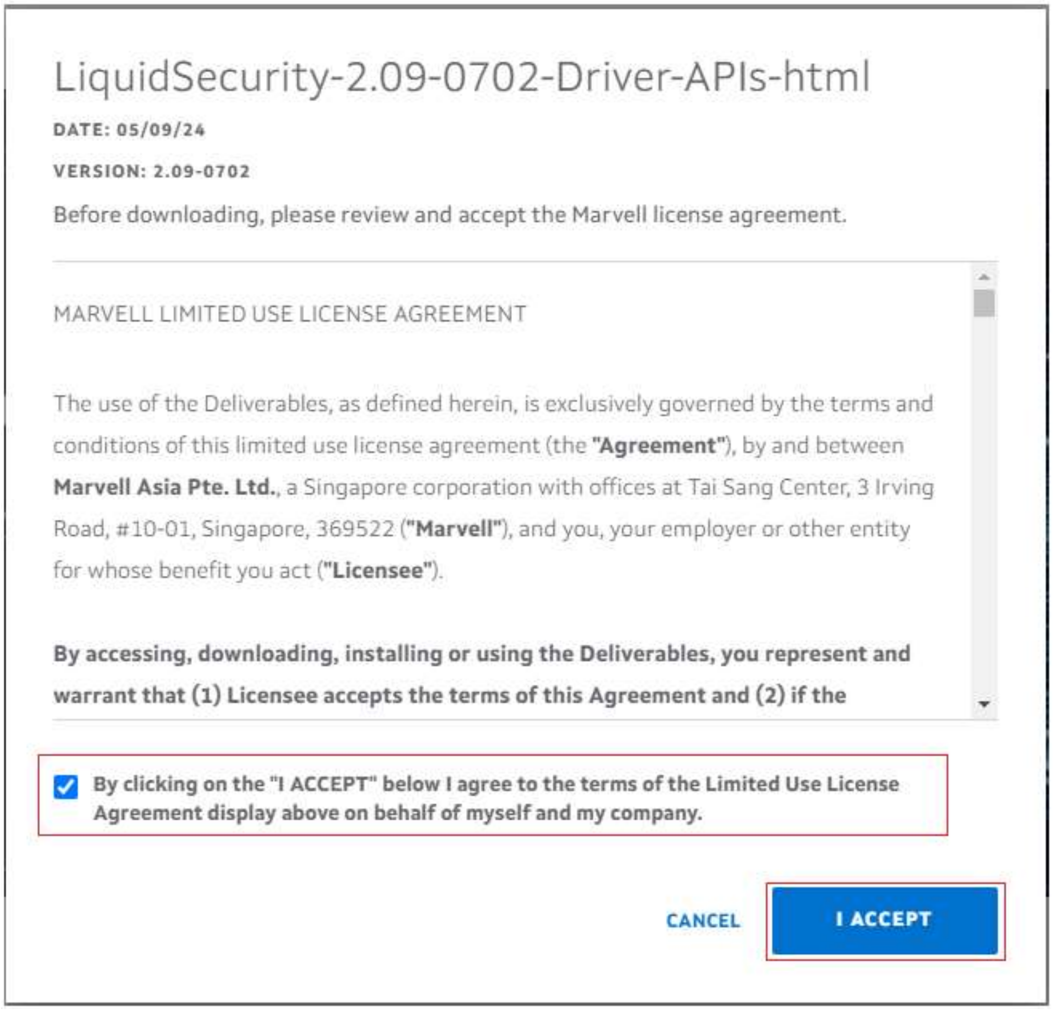
KEYWORDS: Example: RHEL 9.0, Boot code, management tools,documentations, 4.1.57 etc.

2 Results Found

Results per page: 25

DATE	DESCRIPTION	CATEGORY	OS	TYPE	VERSION	DOWNLOAD
05/02/23	**Marvell AQtion Linux driver** Marvell AQtion Linux driver	Marvell Public Drivers	Linux	Drivers	2.5.6	⭳
05/09/24	**LiquidSecurity-2.09-0702-Driver-APIs-html** LiquidSecurity 2.09-0702 Driver APIs	Marvell Public Drivers	Linux	Drivers	2.09-0702	⭳

3. This pops up a window to accept the “MARVELL LIMITED USE LICENSE AGREEMENT”.



LiquidSecurity-2.09-0702-Driver-APIs-html

DATE: 05/09/24

VERSION: 2.09-0702

Before downloading, please review and accept the Marvell license agreement.

MARVELL LIMITED USE LICENSE AGREEMENT

The use of the Deliverables, as defined herein, is exclusively governed by the terms and conditions of this limited use license agreement (the **"Agreement"**), by and between **Marvell Asia Pte. Ltd.**, a Singapore corporation with offices at Tai Sang Center, 3 Irving Road, #10-01, Singapore, 369522 (**"Marvell"**), and you, your employer or other entity for whose benefit you act (**"Licensee"**).

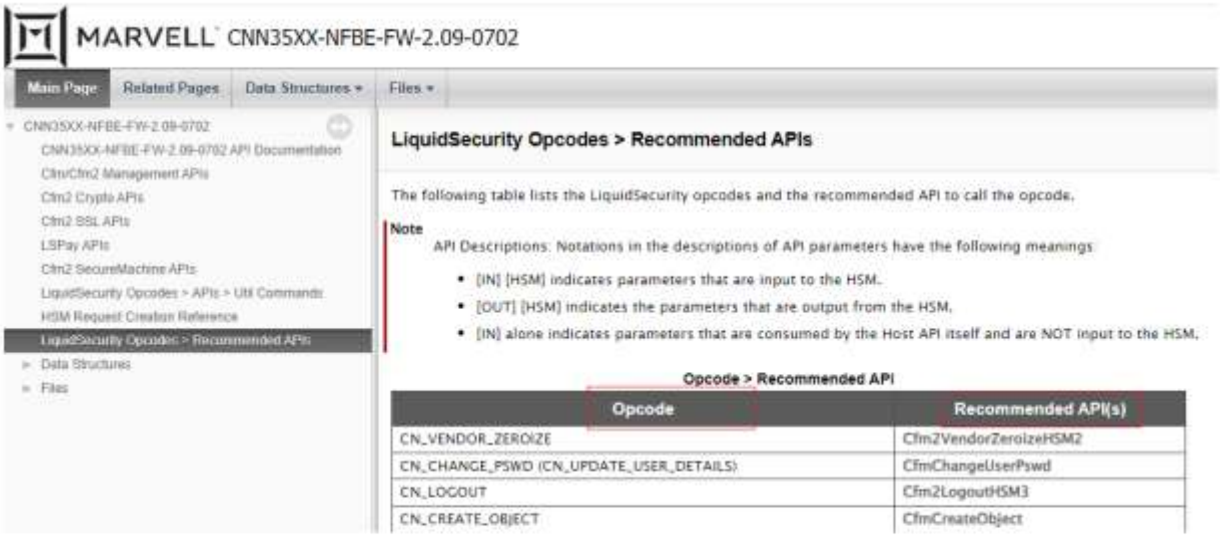
By accessing, downloading, installing or using the Deliverables, you represent and warrant that (1) Licensee accepts the terms of this Agreement and (2) if the

☑ **By clicking on the "I ACCEPT" below I agree to the terms of the Limited Use License Agreement display above on behalf of myself and my company.**

Click **I Accept** to accept the terms and Conditions; the Service Interface document will be downloaded.

4. After the Interface document is downloaded, extract the archive with the Password “LS-FIPS-140-3”.

5. To access the Services, open the `index.html` file; then select **LiquidSecurity Opcodes > Recommended APIs** from the left pane**.** The page depicted in the following snapshot displays:


MARVELL CNN35XX-NFBE-FW-2.09-0702
Main Page
Related Pages
Data Structures ▾
Files ▾
CNN35XX-NFBE-FW-2.09-0702
CNN35XX-NFBE-FW-2.09-0702 API Documentation
Cfm/Cfm2 Management APIs
Cfm2 Crypto APIs
Cfm2 SSL APIs
LSPay APIs
Cfm2 SecureMachine APIs
LiquidSecurity Opcodes > APIs > Util Commands
HSM Request Creation Reference
LiquidSecurity Opcodes > Recommended APIs
Data Structures
Files

LiquidSecurity Opcodes > Recommended APIs

The following table lists the LiquidSecurity opcodes and the recommended API to call the opcode.

Note

API Descriptions: Notations in the descriptions of API parameters have the following meanings

- [IN] [HSM] indicates parameters that are input to the HSM.
- [OUT] [HSM] indicates the parameters that are output from the HSM.
- [IN] alone indicates parameters that are consumed by the Host API itself and are NOT input to the HSM.

Opcode > Recommended API

Opcode	Recommended API(s)
CN_VENDOR_ZEROIZE	Cfm2VendorZeroizeHSM2
CN_CHANGE_PSWD (CN_UPDATE_USER_DETAILS)	CfmChangeUserPswd
CN_LOGOUT	Cfm2LogoutHSM3
CN_CREATE_OBJECT	CfmCreateObject

6. Click the **Recommended API** for each opcode (services) for the input and output details.

Table 10 – Roles, Service Commands, Input and Output

Role	Service	Input	Output
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_ZEROIZE	Opcode inputs	Opcode outputs
MCO/PCO	CN_VENDOR_ZEROIZE	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_APP_INITIALIZE	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_APP_FINALIZE	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_OPEN_SESSION	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_CLOSE_SESSION	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_GET_SESSION_INFO	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_CLOSE_ALL_SESSIONS	Opcode inputs	Opcode outputs
MCO/PCO	CN_CLOSE_PARTITION_SESSIONS	Opcode inputs	Opcode outputs
UN-AUTH	CN_ENCRYPT_SESSION	Opcode inputs	Opcode outputs
UN-AUTH	CN_AUTHORIZE_SESSION	Opcode inputs	Opcode outputs
UN-AUTH	CN_LOGIN	Opcode inputs	Opcode outputs
MCO/PCO/PCU/AU	CN_LOGOUT	Opcode inputs	Opcode outputs
MCO/PCO/PCU/AU	CN_UPDATE_USER_DETAILS	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_TOKEN_INFO	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_PARTITION_INFO	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_GET_HSM_LABEL	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_ALL_PARTITION_INFO	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_GET_POLICY_SET	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_GET_M_VALUE	Opcode inputs	Opcode outputs

Role	Service	Input	Output
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_GET_VERSION	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_GET_CORE_DUMP	Opcode inputs	Opcode outputs
MCO/PCO/PCU/MFG/AU/UN-AUTH	CN_DELETE_CORE_DUMP	Opcode inputs	Opcode outputs
MCO	CN_MASTER_CONFIG	Opcode inputs	Opcode outputs
MCO/PCO/PCU/AU/UN-AUTH	CN_CERT_AUTH_GET_CERT_REQ	Opcode inputs	Opcode outputs
MCO/PCO	CN_CERT_AUTH_STORE_CERT	Opcode inputs	Opcode outputs
MCO/PCO	CN_STORE_VENDOR_PRE_SHARED_KEY (CN_STORE_KBK_SHARE)	Opcode inputs	Opcode outputs
MCO/PCO	CN_SET_KBK_PRIMARY	Opcode inputs	Opcode outputs
MCO/PCO	CN_GET_KBK_SLOT_INFO	Opcode inputs	Opcode outputs
MCO	CN_SHUTDOWN	Opcode inputs	Opcode outputs
MCO/PCO/PCU/AU/UN-AUTH	CN_GET_LOGIN_FAILURE_CNT	Opcode inputs	Opcode outputs
PCO/PCU/AU/UN-AUTH	CN_OPEN_SESSION_V2	Opcode inputs	Opcode outputs
PCO/PCU/AU/UN-AUTH	CN_ENCRYPT_SESSION_V2	Opcode inputs	Opcode outputs
MCO/PCO	CN_INIT_TOKEN	Opcode inputs	Opcode outputs
MCO/PCO	CN_GEN_PSWD_ENC_KEY	Opcode inputs	Opcode outputs
UN-AUTH	CN_UNLOCK_CO	Opcode inputs	Opcode outputs
UN-AUTH	CN_GET_CHALLENGE_CO	Opcode inputs	Opcode outputs
MCO/PCO	CN_INIT_DONE	Opcode inputs	Opcode outputs
MCO/PCO/PCU/AU/UN-AUTH	CN_CERT_AUTH_GET_CERT	Opcode inputs	Opcode outputs
MCO/UN-AUTH	CN_CERT_AUTH_SECURE_BOOT	Opcode inputs	Opcode outputs
MCO	CN_FW_UPDATE_BEGIN CN_FW_UPDATE CN_FW_UPDATE_END	Opcode inputs	Opcode outputs
MCO	CN_SLAVE_CONFIG	Opcode inputs	Opcode outputs
MCO	CN_INVOKE_FIPS	Opcode inputs	Opcode outputs
MCO	CN_GET_RSA_CACHE_SIZE	Opcode inputs	Opcode outputs
MCO/PCO	CN_SET_HSM_CONFIG	Opcode inputs	Opcode outputs
MCO	CN_GET_HSM_DIAG_INFO	Opcode inputs	Opcode outputs
MCO	CN_GET_HSM_WT_PARAM	Opcode inputs	Opcode outputs
MCO	CN_SET_HSM_WT_PARAM	Opcode inputs	Opcode outputs
PCO/PCU/AU/UN-AUTH	CN_GET_SERVER_PARAMS	Opcode inputs	Opcode outputs
MCO	CN_DIAG_GET_HSM_STATS	Opcode inputs	Opcode outputs
MCO	CN_DIAG_GET_PARTITION_STATS	Opcode inputs	Opcode outputs
MCO	CN_STORE_FW_SIGNING_KEY	Opcode inputs	Opcode outputs
MCO	CN_ALLOW_FW_UPDATE	Opcode inputs	Opcode outputs
MCO	CN_GET_BOOT_DATA	Opcode inputs	Opcode outputs

Role	Service	Input	Output
MCO/PCO	CN_SET_CFG_PREGEN_CACHE_SZ	Opcode inputs	Opcode outputs
MCO/PCO	CN_GET_CFG_PREGEN_CACHE_SZ	Opcode inputs	Opcode outputs
MCO/PCO	CN_GET_CFG_PREGEN_CACHE_VAL	Opcode inputs	Opcode outputs
MCO	CN_SET_INIT_TIME	Opcode inputs	Opcode outputs
MCO	CN_SET_VENDOR_TIME	Opcode inputs	Opcode outputs
MCO/UN-AUTH	CN_GET_TIME	Opcode inputs	Opcode outputs
MCO	CN_SYNC_TIME	Opcode inputs	Opcode outputs
MCO	CN_PARTN_STORAGE_GET	Opcode inputs	Opcode outputs
MCO	CN_PARTN_STORAGE_UPDATE	Opcode inputs	Opcode outputs
MCO	CN_PARTN_STORAGE_DELETE	Opcode inputs	Opcode outputs
MCO	CN_CREATE_PARTITION	Opcode inputs	Opcode outputs
MCO	CN_RESIZE_PARTITION	Opcode inputs	Opcode outputs
MCO	CN_DELETE_PARTITION	Opcode inputs	Opcode outputs
MCO/UN-AUTHMCO	CN_GET_PARTITION_COUNT	Opcode inputs	Opcode outputs
MCO/PCO	CN_BACKUP_BEGIN	Opcode inputs	Opcode outputs
MCO/PCO	CN_BACKUP_CONFIG	Opcode inputs	Opcode outputs
MCO/PCO	CN_BACKUP_USERS	Opcode inputs	Opcode outputs
MCO/PCO	CN_BACKUP_KEY	Opcode inputs	Opcode outputs
MCO/PCO	CN_BACKUP_END	Opcode inputs	Opcode outputs
MCO/PCO	CN_RESTORE_BEGIN	Opcode inputs	Opcode outputs
MCO/PCO	CN_RESTORE_CONFIG	Opcode inputs	Opcode outputs
MCO/PCO	CN_RESTORE_USERS	Opcode inputs	Opcode outputs
MCO/PCO	CN_RESTORE_KEY	Opcode inputs	Opcode outputs
MCO/PCO	CN_RESTORE_END	Opcode inputs	Opcode outputs
MCO/PCO	CN_BACKUP_OBJECT	Opcode inputs	Opcode outputs
PCO	CN_WRAP_KBK (Modes: KBK_WRAP_WITH_KEK, KBK_WRAP_WITH_CERT_AUTH_DERIVED_KEY, KBK_WRAP_WITH_RSA)	Opcode inputs	Opcode outputs
PCO	CN_UNWRAP_KBK (Modes: KBK_WRAP_WITH_KEK, KBK_WRAP_WITH_CERT_AUTH_DERIVED_KEY, KBK_WRAP_WITH_RSA)	Opcode inputs	Opcode outputs
MCO/PCO	CN_RESTORE_OBJECT	Opcode inputs	Opcode outputs
PCO	CN_SET_M_VALUE	Opcode inputs	Opcode outputs
PCO/AU	CN_SET_NODEID	Opcode inputs	Opcode outputs
MCO/PCO	CN_SET_POLICY	Opcode inputs	Opcode outputs
PCO	CN_CREATE_USER, CN_CREATE_PRE_OFFICER, CN_CREATE_CO, CN_CREATE_APPLIANCE_USER	Opcode inputs	Opcode outputs

Role	Service	Input	Output
MCO/PCO	CN_DELETE_USER	Opcode inputs	Opcode outputs
PCO/PCU/AU/UN-AUTH	CN_LIST_USERS	Opcode inputs	Opcode outputs
PCO/PCU	CN_GET_USER_INFO	Opcode inputs	Opcode outputs
PCO/MCO	CN_UNLOCK_USER	Opcode inputs	Opcode outputs
PCU	CN_ALWAYS_AUTHORIZE_USER	Opcode inputs	Opcode outputs
PCO	CN_CERT_AUTH_GET_SOURCE_RANDOM	Opcode inputs	Opcode outputs
PCO	CN_CERT_AUTH_VALIDATE_PEER_CERTS	Opcode inputs	Opcode outputs
PCO	CN_CERT_AUTH_SOURCE_KEY_EXCHANGE	Opcode inputs	Opcode outputs
PCO	CN_CLONE_SOURCE_INIT	Opcode inputs	Opcode outputs
PCO	CN_CLONE_SOURCE_STAGE1	Opcode inputs	Opcode outputs
PCO	CN_CLONE_TARGET_INIT	Opcode inputs	Opcode outputs
PCO	CN_CLONE_TARGET_STAGE1	Opcode inputs	Opcode outputs
PCO	CN_CERT_AUTH_TARGET_KEY_EXCHANGE	Opcode inputs	Opcode outputs
PCU	CN_CREATE_OBJECT	Opcode inputs	Opcode outputs
PCO/MCO	CN_GEN_KEY_ENC_KEY	Opcode inputs	Opcode outputs
PCO/PCU/AU	CN_EXTRACT_MASKED_OBJECT	Opcode inputs	Opcode outputs
PCO/PCU/AU	CN_INSERT_MASKED_OBJECT	Opcode inputs	Opcode outputs
PCU	CN_DESTROY_OBJECT	Opcode inputs	Opcode outputs
PCU	CN_GET_ATTRIBUTE_VALUE	Opcode inputs	Opcode outputs
PCU	CN_GET_ATTRIBUTE_SIZE	Opcode inputs	Opcode outputs
PCU	CN_GET_ALL_ATTRIBUTE_SIZE	Opcode inputs	Opcode outputs
PCO/PCU	CN_GET_ALL_ATTRIBUTE_VALUE	Opcode inputs	Opcode outputs
PCO/PCU	CN_MODIFY_OBJECT	Opcode inputs	Opcode outputs
PCO/PCU/AU	CN_FIND_OBJECTS_USING_COUNT/CN_FIND_ALL_OBJECTS_IN_RANGE/CN_FIND_ALL_OBJECTS/CN_FIND_ALL_OBJECTS_USING_COUNT/CN_FIND_OBJECTS/CN_FIND_OBJECTS_FROM_INDEX	Opcode inputs	Opcode outputs
PCU	CN_GENERATE_KEY	Opcode inputs	Opcode outputs
PCU	CN_SPLIT_SECRET_KEY	Opcode inputs	Opcode outputs
PCU	CN_GENERATE_KEY_PAIR	Opcode inputs	Opcode outputs
PCU	CN_EXPORT_PUB_KEY	Opcode inputs	Opcode outputs
PCU	CN_SHARE_OBJECT	Opcode inputs	Opcode outputs
PCU	CN_GET_OBJECT_INFO	Opcode inputs	Opcode outputs
PCU	CN_TOMBSTONE_OBJECT	Opcode inputs	Opcode outputs
PCU	CN_DELETE_TOMBSTONED_OBJECT	Opcode inputs	Opcode outputs
PCU	CN_UNWRAP_KEY/CN_UNWRAP_KEY2	Opcode inputs	Opcode outputs
PCU	CN_WRAP_KEY/CN_WRAP_KEY2	Opcode inputs	Opcode outputs

Role	Service	Input	Output
PCU	CN_NIST_AES_WRAP_UNWRAP/ CN_NIST_AES_WRAP_UNWRAP2	Opcode inputs	Opcode outputs
MCO	CN_GET_RSA_CACHE_SIZE	Opcode inputs	Opcode outputs
PCU	CN_DERIVE_KEY	Opcode inputs	Opcode outputs
PCO	CN_MODIFY_KEY_OWNER	Opcode inputs	Opcode outputs
MCO/PCO/PCU/AU	CN_ADMIN_GET_PARTN_KEYHANDLES_HASH	Opcode inputs	Opcode outputs
PCO/AU	CN_GET_PARTN_SINGLE_KEYHANDLE_HASH	Opcode inputs	Opcode outputs
PCU	CN_PARK_OBJECT	Opcode inputs	Opcode outputs
PCU	CN_UNPARK_OBJECT	Opcode inputs	Opcode outputs
PCO	CN_SET_USER_ATTR	Opcode inputs	Opcode outputs
PCO	CN_LIST_AUTH_PUB_KEYS	Opcode inputs	Opcode outputs
PCO	CN_CERT_AUTH_REMOVE_CERT	Opcode inputs	Opcode outputs
PCO/AU	CN_PARTN_GET_AUDIT_DETAILS	Opcode inputs	Opcode outputs
PCO/AU	CN_PARTN_GET_AUDIT_LOGS	Opcode inputs	Opcode outputs
PCO/AU	CN_PARTN_GET_AUDIT_SIGN	Opcode inputs	Opcode outputs
PCO/AU	CN_PARTN_ACK_AUDIT_SIGN	Opcode inputs	Opcode outputs
MCO	CN_FINALIZE_LOGS	Opcode inputs	Opcode outputs
PCU	CN_SIGN	Opcode inputs	Opcode outputs
PCU	CN_VERIFY	Opcode inputs	Opcode outputs
PCU	CN_ECC_DH	Opcode inputs	Opcode outputs
PCU	CN_NIST_AES_WRAP	Opcode inputs	Opcode outputs
PCU	CN_ALLOC_SSL_CTX	Opcode inputs	Opcode outputs
PCU	CN_FREE_SSL_CTX	Opcode inputs	Opcode outputs
PCU	CN_GEN_PMK	Opcode inputs	Opcode outputs
PCU	CN_FIPS_RAND	Opcode inputs	Opcode outputs
PCU	CN_ME_PKCS_LARGE	Opcode inputs	Opcode outputs
PCU	CN_ME_PKCS	Opcode inputs	Opcode outputs
PCU	CN_FECC	Opcode inputs	Opcode outputs
PCU	CN_HASH	Opcode inputs	Opcode outputs
PCU	CN_HMAC	Opcode inputs	Opcode outputs
PCU	CN_ENCRYPT_DECRYPT	Opcode inputs	Opcode outputs
PCU	MAJOR_OP_OTHER	Opcode inputs	Opcode outputs
PCU	MAJOR_OP_FINISHED	Opcode inputs	Opcode outputs
PCU	MAJOR_OP_RESUME	Opcode inputs	Opcode outputs
PCU	MAJOR_OP_ENCRYPT_DECRYPT_RECORD	Opcode inputs	Opcode outputs
PCU	CN_SHA3	Opcode inputs	Opcode outputs

Role	Service	Input	Output
PCU	MAJOR_OP_DECRYPT_AND_ENCRYPT	Opcode inputs	Opcode outputs
PCO/PCU	CN_GET_TOKEN	Opcode inputs	Opcode outputs
PCO/PCU	CN_APPROVE_TOKEN	Opcode inputs	Opcode outputs
PCO/PCU	CN_LIST_TOKENS	Opcode inputs	Opcode outputs
PCO	CN_TOKEN_TIMEOUT	Opcode inputs	Opcode outputs
PCO/PCU	CN_DELETE_TOKEN	Opcode inputs	Opcode outputs
MCO	CN_SM_IMAGE_DELETE	Opcode inputs	Opcode outputs
MCO	CN_SME_DIAG_INFO	Opcode inputs	Opcode outputs
MCO	CN_SET_SM_APP_CONFIG	Opcode inputs	Opcode outputs
MCO	CN_GET_SM_APP_CONFIG	Opcode inputs	Opcode outputs
MCO	CN_SET_SM_CAPABILITY	Opcode inputs	Opcode outputs
MCO	CN_GET_SM_CONFIG	Opcode inputs	Opcode outputs
PCO	CN_SMAPP_DIAG_INFO	Opcode inputs	Opcode outputs
PCO	CN_SMAPP_UPDATE_BEGIN	Opcode inputs	Opcode outputs
PCO	CN_SMAPP_UPDATE	Opcode inputs	Opcode outputs
PCO	CN_SMAPP_UPDATE_END	Opcode inputs	Opcode outputs
PCO	CN_SMAPP_CTRL	Opcode inputs	Opcode outputs
PCO	CN_SMAPP_DELETE	Opcode inputs	Opcode outputs
PCU	CN_SMAPP_WRITE_DATA	Opcode inputs	Opcode outputs
PCU	CN_SMAPP_READ_DATA	Opcode inputs	Opcode outputs
PCU	CN_SMAPP_DELETE_FILE	Opcode inputs	Opcode outputs
MCO	CN_SET_SM_CONFIG	Opcode inputs	Opcode outputs
PCU	CN_SMAPP_WRITE_SFRAM	Opcode inputs	Opcode outputs
PCU	CN_SMAPP_READ_SFRAM	Opcode inputs	Opcode outputs
PCO	CN_LIST_UNLINKED_OBJECTS	Opcode inputs	Opcode outputs
PCO/PCU	CN_GENERATE_PBE_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_GENERATE_ASYMM_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_GENERATE_SYMM_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_EXPORT_PUBLIC_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_IMPORT_PUBLIC_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_VALIDATE_PUBLIC_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_IMPORT_KPK	Opcode inputs	Opcode outputs
PCU	LSPAY_EXPORT_KPK	Opcode inputs	Opcode outputs
PCU	LSPAY_IMPORT_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_IMPORT_TR34_KEY	Opcode inputs	Opcode outputs

Role	Service	Input	Output
PCU	LSPAY_EXPORT_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_EXPORT_TR34_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_TRANSLATE_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_IMPORT_CERT	Opcode inputs	Opcode outputs
PCU	LSPAY_IMPORT_DECIMAL_TABLE	Opcode inputs	Opcode outputs
PCU	LSPAY_GENERATE_CSR	Opcode inputs	Opcode outputs
PCU	LSPAY_DERIVE_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_ENCRYPT	Opcode inputs	Opcode outputs
PCU	LSPAY_DECRYPT	Opcode inputs	Opcode outputs
PCU	LSPAY_DECRYPT_THEN_ENCRYPT	Opcode inputs	Opcode outputs
PCU	LSPAY_MAC_GEN	Opcode inputs	Opcode outputs
PCU	LSPAY_MAC_VERIFY	Opcode inputs	Opcode outputs*
PCU	LSPAY_MAC_TRANSLATE	Opcode inputs	Opcode outputs*
PCU	LSPAY_FPE_ENCRYPT	Opcode inputs	Opcode outputs
PCU	LSPAY_FPE_DECRYPT	Opcode inputs	Opcode inputs
PCU	LSPAY_SIGN	Opcode inputs	Opcode outputs
PCU	LSPAY_SIGN_VERIFY	Opcode inputs	Opcode inputs
PCU	LSPAY_PIN_BLOCK_TRANSLATE	Opcode inputs	Opcode outputs
PCU	LSPAY_DERIVE_PIN_FROM_OFFSET	Opcode inputs	Opcode outputs
PCU	LSPAY_DERIVE_OFFSET_FROM_PIN	Opcode inputs	Opcode outputs
PCU	LSPAY_VERIFY_PIN	Opcode inputs	Opcode outputs
PCU	LSPAY_PVV_GENERATION	Opcode inputs	Opcode outputs
PCU	LSPAY_PVV_VERIFY	Opcode inputs	Opcode outputs
PCU	LSPAY_EMV_GENVERIFY_AC	Opcode inputs	Opcode outputs
PCU	LSPAY_EMV_SECURE_MSG_GEN	Opcode inputs	Opcode outputs
PCU	LSPAY_CVV_GEN	Opcode inputs	Opcode outputs
PCU	LSPAY_CVV_VERIFY	Opcode inputs	Opcode outputs
PCU	LSPAY_KEY_SHARE_CREATE	Opcode inputs	Opcode outputs
PCU	LSPAY_KEY_SHARE_EXPORT_KEY_COMPONENT	Opcode inputs	Opcode outputs
PCU	LSPAY_KEY_SHARE_COMBINE_INIT	Opcode inputs	Opcode outputs
PCU	LSPAY_KEY_SHARE_IMPORT_COMPONENT	Opcode inputs	Opcode outputs
PCU	LSPAY_KEY_SHARE_COMBINE_KEY	Opcode inputs	Opcode outputs
PCU	LSPAY_KEY_SHARE_ZEROIZE	Opcode inputs	Opcode outputs
PCO	LSPAY_MFK_GENERATE	Opcode inputs	Opcode outputs
PCO	LSPAY_MFK_GET_INFO	Opcode inputs	Opcode outputs

Role	Service	Input	Output
PCO	LSPAY_MFK_DELETE	Opcode inputs	Opcode outputs
PCO	LSPAY_MFK_SET_PRIMARY	Opcode inputs	Opcode outputs
PCU	MAJOR_OP_RSASERVER_LARGE	Opcode inputs	Opcode outputs
PCU	MAJOR_OP_RSASERVER	Opcode inputs	Opcode outputs
UN-AUTH	Firmware Update	Serial inputs: run boot_usb	RED LED BLINK
PCU	CN_GENERATE_KEY_PAIR (non-compliant)	Opcode inputs	Opcode outputs
PCU	CN_GENERATE_KEY (non-compliant)	Opcode inputs	Opcode outputs
PCU	CN_CREATE_OBJECT (non-compliant)	Opcode inputs	Opcode outputs
PCU	CN_UNWRAP_KEY (non-compliant)	Opcode inputs	Opcode outputs
PCU	CN_WRAP_KEY (non-compliant)	Opcode inputs	Opcode outputs
PCU/PCO/AU	CN_EXTRACT_MASKED_OBJECT (non-compliant)	Opcode inputs	Opcode outputs
MCO	CN_STORE_FW_SIGNING_KEY (non-compliant)	Opcode inputs	Opcode outputs
PCU	CN_ME_PKCS_LARGE (non-compliant) CN_ME_PKCS (non-compliant)	Opcode inputs	Opcode outputs
Manufacturer	CN_STORE_VENDOR_PRE_SHARED_KEY (CN_STORE_KBK_SHARE) (non-compliant)	Opcode inputs	Opcode outputs
PCU/PCO/AU	CN_INSERT_MASKED_OBJECT (non-compliant)	Opcode inputs	Opcode outputs
UN-AUTH	CN_ENCRYPT_SESSION (non-compliant)	Opcode inputs	Opcode outputs
PCU	CN_DERIVE_KEY (non-compliant)	Opcode inputs	Opcode outputs

- CN_INSERT_MASKED_OBJECT includes the opcode CN_INSERT_MASKED_OBJECT_USER.
- CN_CERT_AUTH_GET_CERT includes the opcode CN_CERT_AUTH_GET_CERT_CHAIN.
- CN_CERT_AUTH_STORE_CERT includes the opcode CN_CERT_AUTH_STORE_CERT_CHAIN.

4.1 Assumption of Roles, Master Partition

The module supports the following roles. One identity is allowed for each role, per partition.

4.1.1 Manufacturer (MFG)

During the manufacturing stage, each HSM goes through the following process:

- An RSA key pair called the HSM FIPS Master Authentication Key (FMAK) is generated on HSM. CSR is requested out of HSM and signed by the Manufacturer Authentication Root Certificate (MARC). The generated certificate is called the HSM FIPS Master Authentication Certificate (FMAC).
- A 256-bit MKBK encrypted with the FMAK public key is loaded into the HSM.
- Program Performance settings and capabilities Appliance Compatibility mode, run random operations, Encrypted channels.
- Program Serial Number and Max Operating Temperature

The same above steps are followed by the manufacturer once the HSM is moved to manufacturer reset after manufacturer zeroize.

4.1.2 Master Crypto Officer (MCO)

The master partition supports only the Master Crypto Officer role (MCO). This role is used to configure non-master partitions (create, provision, resize, delete) but cannot access their resources (e.g., cannot manage or use non-master partition keys).

This role is authenticated with username and password (one-factor) and optionally with signature as well (two-factor). Refer to Section 4.3 for details.

4.2 Assumption of Roles, Non-Master Partition Roles

Each Non-Master Partition supports four (4) distinct operator roles as described below. The module enforces the separation of roles using identity-based authentication. Re-authentication is required to change roles.

Except for Pre-CO, concurrent operators are allowed; however, only one operator is allowed per login session.

4.2.1 Pre-Crypto Officer (Pre-CO)

During partition initialization, default credentials are used to create a Pre-CO or a PCO. The Pre-CO is a restricted role primarily for configuring certificates and setting up a PCO. Once a PCO is set up for a partition, the Pre-CO role is no longer accessible.

Because the Pre-CO is essentially a restricted PCO, it does not have its own column in Table 12. Instead, PCO capabilities in Table 12 are marked with an asterisk (*) to indicate Pre-CO can run these services.

This role is authenticated with username and password (one-factor) only.

4.2.2 Partition Crypto Officer (PCO)

This role has access to administrative services of the partition and can configure PCU and AU identities.

This role is authenticated with username and password (one-factor) and optionally with signature as well (two-factor).

4.2.3 Partition Crypto User (PCU)

This role has access to all cryptographic services offered by the partition; its purpose is operational use of the module.

This role is authenticated with username and password (one-factor) and optionally with signature as well (two-factor).

4.2.4 Appliance User (AU)

This role has access to partition audit logs and can create end-to-end encrypted channels. It is to set up and synchronize clusters.

This role is authenticated with username and password (one-factor) only.

4.3 Authentication

The module enforces identity-based authentication. A role is explicitly selected at authentication. If a given identity is not allocated with the role, then the authentication will fail with appropriate error. The MCO role is associated with the Master Partition, and the PCO and PCU roles are associated with user partitions (see Sections 4.41 and 4.2 for details). If the given user identity is not allocated for a role, then authentication will be failed with appropriate error. The module allows one identity per role, per partition.

All the user roles should be authenticated with Level 3 based authentication mechanisms. MCO, Pre-CO, PCO, and PCU should be authenticated using Memorized-Secret and optionally single-factor crypto software (2FA). The MFG role is accepted only with single-factor crypto software. Identity is determined by certificate for MFG role and with username for all other roles.

Table 11 – Roles and Required Identification and Authentication

Role	Description	Authentication Type	Authentication Data
MFG	This role sets the identity, serial number, performance settings and max operating temperature	Manufacturer License certificate-based authentication	RSA signature [single factor crypto software]
MCO	This role has access to administrative services offered by the module or HSM	Identity-based operator authentication	Username and password [Memorized-Secret]; optional RSA signature (2FA) [single factor crypto software]
Pre-CO	This role is an optional role with limited functionality, eventually transition into PCO	Identity-based operator authentication	Username and password [Memorized-Secret]
PCO	This role has access to administrative services of the partition	Identity-based operator authentication	Username and password [Memorized-Secret]; optional RSA signature (2FA) [single factor crypto software]
PCU	This role has access to all crypto services offered by the partition	Identity-based operator authentication	Username and password Memorized-Secret]; optional RSA signature (2FA) [single factor crypto software]
AU	This role has access to partition audit logs and Appliance secure channel key	Identity-based operator authentication	Username and password Memorized-Secret]

Table 12 – Roles and Authentication

Role	Authentication Method	Authentication Strength
MCO/Pre-CO/PCO/PCU/AU	Username and password	The password is a minimum of 8 characters, case-sensitive alpha-numeric. As such there are (26*2+10) ^8 = 62^8 possible minimum-length passwords, and the false acceptance rate is 1 in 62^8. A maximum of 20 password attempts are possible before permanent lockout. Therefore, the probability of false authentication over any timeframe is 20 in 62^8. (The number of allowed login attempts prior to lockout is configured during module initialization but cannot exceed 20.) Lockout of MCO automatically zeroizes the module. In all other cases, lockout can be unset by destroying the partition.
MCO/PCO/PCU MFG	RSA Signature	Authentication is performed using SHA2-256 based RSA 2048-bit PKCS#1-v1.5 signatures (provides 112 bits of strength). Corresponding public key is associated with the identity (for Manufacturing role, it is part of FW image). The probability that a random attempt will succeed, or a false acceptance will occur, is approximately 1 in 2^112, which is less than 1 in 1,000,000. For each failed signature verification, the module will block for 2 seconds. Based on this maximum rate, the probability that a random attempt will succeed in a one-minute period is approximately 30 in 2^112, which is less than 1 in 100,000.

4.4 Roles, Services, and CSP Access

G = Generate: The module generates or derives the SSP.

R = Read: The SSP is read from the module (e.g., the SSP is output).

W = Write: The SSP metadata is updated. The module writes the SSP. The write access is typically performed after a SSP is imported into the module, or the module generates an SSP, or the module overwrites an existing SSP.

E = Execute: The module uses the SSP in performing a cryptographic operation.

Z = Zeroize: The module zeroizes the SSP.

Table 13 – Approved Services

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_ZEROIZE	• Zeroize the HSM Master/user partition. Can be configured to be allowed by CO only • Master zeroize will zeroize and delete all user partitions • With factory_reset option, all the SSPs of the partition will be zeroized. • Regular zeroize will zeroize all the user(s) generated SSPs of the partition • Please refer to Table 18 for the list of SSPs and corresponding zeroization service types that erase them.	None	User keys MMEK PMEK PAK KLK Partition Masking Key PAC 2FAMofNPubK CAPubK AOAPubK Login Passwords PEK KBK POTAC POKBK POAC	MCO/PCO/ PCU/MFG/AU/UN-AUTH	Z	Success with fips_state = 2 or 3
CN_VENDOR_ZEROIZE	Zeroizes HSM Master partition. • Vendor zeroize (MCO only) zeroizes vendor programmed certificates and SSPs. factory reset will zeroize all the SSPs of the partition. Please refer to Table 18 for the list of SSPs which can be zeroized using the current zeroization method.	None	User keys MMEK PMEK PAK KLK Partition Masking Key PAC 2FAMofNPubK CAPubK AOAPubK Login Passwords PEK KBK POTAC POKBK POAC AOTAC OKBK AOAC SecureBootAuth Public Key On vendor zeroize: MFDEK FMAK MFKBK	MCO/PCO	Z	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
			Manufacturer Firmware Integrity Check Keys MARC FMAC			
CN_APP_INITIALIZE	Registers an application with HSM.	Counter DRBG (SP 800-90Ar1) [#C821]	DRBG ENTROPY/CTR_DRBG Internal State	MCO/PCO/PCU/MFG/AU/UN-AUTH	E R	Success with fips_state = 2 or 3
CN_APP_FINALIZE	Unregisters an application from HSM.	None	User keys	MCO/PCO/PCU/MFG/AU/UN-AUTH	Z	Success with fips_state = 2 or 3
CN_OPEN_SESSION	Opens a session in HSM and returns the session handle.	Counter DRBG (SP 800-90Ar1) [#C821]	DRBG ENTROPY/CTR_DRBG Internal State	MCO/PCO/PCU/MFG/AU/UN-AUTH	None	Success with fips_state = 2 or 3
CN_CLOSE_SESSION	Closes the session.	None	User keys	MCO/PCO/PCU/MFG/AU/UN-AUTH	Z	Success with fips_state = 2 or 3
CN_GET_SESSION_INFO	Gets the session information.	None	None	MCO/PCO/PCU/MFG/AU/UN-AUTH	None	Success with fips_state = 2 or 3
CN_CLOSE_ALL_SESSIONS	Management services for closing all sessions of an application.	None	User keys	MCO/PCO/PCU/MFG/AU/UN-AUTH	Z	Success with fips_state = 2 or 3
CN_CLOSE_PARTITION_SESSIONS	Close sessions of all Applications tied to a Partition.	None	User keys	PCO/MCO	Z	Success with fips_state = 2 or 3
CN_ENCRYPT_SESSION	Enables encrypted communication channel.	Hash DRBG (SP 800-90Ar1) [#C830] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] KDF TLS (CVL) (SP 800-135r1) [#C840] KAS-ECC-SSC Sp800-56Ar3 [#A2161]	POAC PAK TLS ECDH Session Key TLS Session Symmetric Key Set TLS Session HMAC Key DRBG Entropy/HASH_DRBGInternal State	UN-AUTH	R E E G G E	Success with fips_state = 2 or 3
CN_AUTHORIZE_SESSION	Authorizes the sessions to be used under E2E and do login.	None	None	UN-AUTH	None	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_LOGIN	Allows login to a session. Public key is used to verify user signatures, optionally in 2-factor authentication.	AES [#C819], allowed per IG D.G RSA SigVer (FIPS 186-4) [#C824] PBKDF (SP 800-132) [#A1196] Counter DRBG (SP 800-90Ar1) [#C821]	PEK Login passwords 2FAMofNPubK CAPubK DRBG Entropy/CTR_DRBG Internal State	UN-AUTH	E W E E	Success with fips_state = 2 or 3
CN_LOGOUT	Allows logout of a session.	None	None	MCO/PCO/PCU/AU	None	Success with fips_state = 2 or 3
CN_UPDATE_USER_DETAILS	Requires user to be logged in. Updates Passwords and Public key for 2-factor authentication or updates the username of PCO.	RSA SigVer (FIPS 186-4) [#C824] AES-CBC (SP 800-38A) [#C839] Counter DRBG (SP 800-90Ar1) [#C821] PBKDF (SP 800-132) [#A1196]	2FAMofNPubK PEK Login passwords DRBG ENTROPY/CTR_DRBG Internal State	MCO/PCO/PCU/AU	W R W E	Success with fips_state = 2 or 3
CN_TOKEN_INFO	Get token information.	None	MFKBK OKBK	MCO/PCO/PCU/MFG/AU/UN-AUTH	R R	Success with fips_state = 2 or 3
CN_PARTITION_INFO	Returns Partition Information.	None	Partition Owner KBK (POKBK)	MCO/PCO/PCU/MFG/AU/UN-AUTH	E	Success with fips_state = 2 or 3
CN_GET_HSM_LABEL	Returns HSM label.	None	None	MCO/PCO/PCU/MFG/AU/UN-AUTH	None	Success with fips_state = 2 or 3
CN_ALL_PARTITION_INFO	Get information for all Partitions.	None	Partition Owner KBK (POKBK)	MCO/PCO/PCU/MFG/AU/UN-AUTH	E	Success with fips_state = 2 or 3
CN_GET_POLICY_SET	Get the current policy settings. This operation does not need authentication.	None	None	MCO/PCO/PCU/MFG/AU/UN-AUTH	None	Success with fips_state = 2 or 3
CN_GET_M_VALUE	Get the current M Value of a CO Service.	None	None	MCO/PCO/PCU/MFG/AU/UN-AUTH	None	Success with fips_state = 2 or 3
CN_GET_VERSION	Obtain Firmware Version.	None	None	MCO/PCO/PCU/MFG/AU/UN-AUTH	None	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_GET_CORE_DUMP	Retrieves core dump files from HSM and saves as dump_file.	None	None	MCO/PCO/PCU/MFG/AU/UN-AUTH	None	Success with fips_state = 2 or 3
CN_DELETE_CORE_DUMP	Delete core dump file if it exists.	None	None	MCO/PCO/PCU/MFG/AU/UN-AUTH	None	Success with fips_state = 2 or 3
CN_MASTER_CONFIG	Perform a master configuration.	RSA SigVer (FIPS 186-4) [#C824]	MLVK MARC FMAK FMAC	MCO	R W G G	Success with fips_state = 2 or 3
CN_CERT_AUTH_GET_CERT_REQ	Get the partition or HSM Certificate Signing Request (CSR).	None	AOAC POAC	MCO/PCO/PCU/AU/UN-AUTH	R R	Success with fips_state = 2 or 3
CN_CERT_AUTH_STORE_CERT	Store the partition owner certificate or the partition certificate signed by the partition owner or HSM certificate signed by vendor, HSM owner certificate and HSM certificate signed by HSM owner.	RSA SigVer (FIPS 186-4) [#C824]	AOTAC POTAC AOAC POAC	MCO/PCO	W W W W	Success with fips_state = 2 or 3
CN_STORE_VENDOR_PRE_SHARED_KEY (CN_STORE_KBK_SHARE)	Store Fixed Keys (KBK) for backup.	RSA SigVer (FIPS 186-4) [#C824] KTS-IFC (KTS) (SP 800-56Br2) [#A1194], AES-KWP (KTS) (SP 800-38F) [#C1263]	MARC AOTAC, POTAC, MFKBK, OKBK, POKBK, FMAK, PAK	MCO/PCO	E E E W W W E E	Success with fips_state = 2 or 3
CN_SET_KBK_PRIMARY	Set the latest stored fixed (KBK) key as the primary key for backup.	None	None	MCO/PCO	None	Success with fips_state = 2 or 3
CN_GET_KBK_SLOT_INFO	Get the stored fixed (KBK) keys ekcv information.	None	None	MCO/PCO	None	Success with fips_state = 2 or 3
CN_SHUTDOWN	Set HSM state to shutdown.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_GET_LOGIN_FAILURE_CNT	Get the login failure count of a particular user.	None	None	MCO/PCO/PCU/AU/UN-AUTH	None	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_OPEN_SESSION_V2	Opens a session in HSM and returns the session handle.	None	None	PCO/PCU/ AU/UN-AUTH	None	Success with fips_state = 2 or 3
CN_ENCRYPT_SESSION_V2	Establishes E2E connection with/without client-authentication, between HSM and the Host applications.	Hash DRBG (SP 800-90Ar1) [#C830] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] KAS TLS (SP 800-56Ar3) KAS-ECC-SSC Sp800-56Ar3 [#A2161] RSA SigVer (FIPS 186-4) [#C824]	E2E Client Authentication Public key PAK POAC E2E TLS ECDH Session Key E2E TLS Session Symmetric Key Set E2E TLS Session HMAC Keys DRBG Entropy/HASH_DRBG Internal State	PCO/PCU/ AU/UN-AUTH	E E R G G G E	Success with fips_state = 2 or 3
CN_INIT_TOKEN	Initializes the HSM and sets its policies and boundaries to the values specified in config_file.	AES-CBC (SP 800-38A) [#C839] RSA SigGen (FIPS 186-4) [#C824] Counter DRBG (SP 800-90Ar1) [#C821] CKG SP 800-133r2 [Vendor affirmed]	MMEK PMEK DRBG ENTROPY/CTR_DRBG Internal State	MCO/PCO	G G E	Success with fips_state = 2 or 3
CN_GEN_PSWD_ENC_KEY	Generates a Password Encryption Key (PEK), which is used to wrap the user password while sending it over the FIPS boundary.	KAS-IFC HKDF (SP 800-56Br2) KAS-IFC OneStep (SP 800-56Br2) KAS-IFC TwoStep (SP 800-56Br2)Counter DRBG (SP 800-90Ar1) [#C821]	PEK Host PswdEncKeyPublic Key DRBG ENTROPY/CTR_DRBG Internal State	MCO/PCO	G E E	Success with fips_state = 2 or 3
CN_UNLOCK_CO	On providing response(signature) over the challenge thrown by HSM/Partition during CN_GET_CHALLENGE_CO (CfmGetChallengeCO), session will be marked with "unlock" privileges, allowing the user to generate PEK, zeroizeHSM and change the CO's self password. Session will remain in unlocked state only for 120 seconds.	RSA SigVer (FIPS 186-4) [#C824]	POAC/AOAC	UN-AUTH	E	Success with fips_state = 2 or 3
CN_GET_CHALLENGE_CO	Gets a challenge to be signed by either HSM/Partition's owner to move the session to "unlocked" state using "unlockco" command.	Counter DRBG (SP 800-90Ar1) [#C821] RSA SigGen (FIPS 186-4) [#C824]	POAC AOAC PAK FMAC	UN-AUTH	R R E R	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
			FMAK DRBG ENTROPY/CTR_DRBG Internal State		E E	
CN_INIT_DONE	Completes initialization of HSM/partition. Successful initialization of HSM will reboot the HSM.	Counter DRBG (SP 800-90Ar1) [#C821] CKG SP 800-133r2 [Vendor affirmed]	KBK Partition masking key DRBG ENTROPY/CTR_DRBG Internal State	MCO/PCO	G G E	Success with fips_state = 2 or 3
CN_CERT_AUTH_GET_CERT	Fetches certificates stored on the HSM. Certificates like: VENDOR_CERT HSM_CERT PARTITION_OWNER_CERT PARTITION_CERT PARTITION_CERT_ISSUED_BY_HSM. HSM_OWNER_CERT HSM_CERT_ISSUED_BY_HO	None	MARC FMAC POAC POTAC PAC AOAC AOTAC	MCO/PCO/PCU/AU/UN-AUTH	R R R R R R R	Success with fips_state = 2 or 3
CN_CERT_AUTH_SECURE_BOOT	Performs cert auth based secure boot.	RSA SigVer (FIPS 186-4) [#C824]	User Public Keys	MCO/UN-AUTH	E	Success with fips_state = 2 or 3
CN_FW_UPDATE_BEGIN CN_FW_UPDATE CN_FW_UPDATE_END	Begins and performs firmware, bootloader, SMW update. Updated FW version is reflected after reboot and can be obtained from the getHSMInfo.	RSA SigVer (FIPS 186-4) [#C824]	MFUVK AOAPubK	MCO	E E	Success with fips_state = 2 or 3
CN_SLAVE_CONFIG	Perform a slave configuration.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_INVOKE_FIPS	Perform Self tests.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_GET_RSA_CACHE_SIZE	Gets the number of RSA pre generated keys available in the RSA key cache.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_SET_HSM_CONFIG	Sets the HSM configuration parameters.	None	None	MCO/ PCO	None	Success with fips_state = 2 or 3
CN_GET_HSM_DIAG_INFO	Get HSM diagnostics information.	None	None	MCO	None	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_GET_HSM_WT_PARAM	Gets the DoS parameter.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_SET_HSM_WT_PARAM	Sets the DoS parameter.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_GET_SERVER_PARAMS	Gets the server parameters used in Cav-server for the server handshake messages.	Counter DRBG (SP 800-90Ar1) [#C821] ECDSA SigVer (FIPS 186-4) [#C825], ECDSA SigVer (FIPS 186-4) [#C829]	TLS Session ECDH Key POAC PAK DRBG ENTROPY/CTR_DRBG Internal State	PCO/PCU/AU/UN-AUTH	G E E E	Success with fips_state = 2 or 3
CN_DIAG_GET_HSM_STATS	Retrieve HSM statistics over fast path.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_DIAG_GET_PARTITION_STATS	Retrieve Partition statistics over fast path.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_STORE_FW_SIGNING_KEY	Configure an RSA or EC public key into HSM as AO attestation key. These keys can be of modulus 1024, 2048, 3072 and 4096 or a supported EC curve.	RSA SigVer (FIPS 186-4) [#C824]	AOAPubK, AOAC	MCO	W E	Success with fips_state = 2 or 3
CN_ALLOW_FW_UPDATE	Configure a lower version of FW to be allowed to be updated for certain time period and on certain HSMs.	RSA SigVer (FIPS 186-4) [#C824] ECDSA SigVer (FIPS 186-4) [#C825]	AOAPubK	MCO	E	Success with fips_state = 2 or 3
CN_GET_BOOT_DATA	Retrieves error logs from HSM.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_SET_CFG_PREGEN_CACHE_SZ	Set pre generated keys cache size.	None	None	MCO/PCO	None	Success with fips_state = 2 or 3
CN_GET_CFG_PREGEN_CACHE_SZ	Set configured pre generated keys cache size.	None	None	MCO/PCO	None	Success with fips_state = 2 or 3
CN_GET_CFG_PREGEN_CACHE_VAL	Returns the key count in pre generated key cache.	None	None	MCO/PCO	None	Success with fips_state = 2 or 3
CN_SET_INIT_TIME	Sets the user's initial time upon receiving the HSM.	None	None	MCO	None	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_SET_VENDOR_TIME	Sets the vendor time on the HSM.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_GET_TIME	Gets the RTC and System time from HSM.	None	None	MCO/ UN-AUTH	None	Success with fips_state = 2 or 3
CN_SYNC_TIME	Sets the user's time on the HSM. Also used for drift calculation and configuration. Returns useful information such as the drift between previous configured time and new time configured and the lifetime average drift observed.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_PARTN_STORAGE_GET	Gets the partition private data from the per partition store.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_PARTN_STORAGE_UPDATE	Updates partition private data into the partition store.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_PARTN_STORAGE_DELETE	Deletes the partition private data from the partition store.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_CREATE_PARTITION	Create a partition with the given name and size.	RSA KeyGen (FIPS 186-4) [#C824] RSA SigGen (FIPS 186-4) [#C824]	PAK FMAC FMAK MARC PAC	MCO	G E E E G	Success with fips_state = 2 or 3
CN_RESIZE_PARTITION	Resize an existing partition of the specified name.	None	N/A	MCO	None	Success with fips_state = 2 or 3
CN_DELETE_PARTITION	Delete a Partition & all associated keys.	None	User Keys PMEK PAK KLK Partition Masking Key PAC 2FAMofNPubK CAPubK Login Passwords PEK KBK POTAC POKBK	MCO	Z	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
			POAC			
CN_GET_PARTITION_COUNT	Return partition count.	None	None	MCO/UN-AUTH	None	Success with fips_state = 2 or 3
CN_BACKUP_BEGIN	Initiate backup of partition configuration, users, and keys.	KDF SP 800-108 (KBKDF) [#C826] Counter DRBG (SP 800-90Ar1) [#C821] HMAC-SHA-1 (FIPS 198-1) [#C822], HMAC-SHA2-224 (FIPS 198-1) [#C822], HMAC-SHA2-256 (FIPS 198-1) [#C822], HMAC-SHA2-384 (FIPS 198-1) [#C822], HMAC-SHA2-512 (FIPS 198-1) [#C822] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	MFKBK OKBK POKBK Backup session Key DRBG ENTROPY/CTR_DRBG Internal State	MCO/PCO	E E E G E	Success with fips_state = 2 or 3
CN_BACKUP_CONFIG	Back up the partition configuration.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] Counter DRBG (SP 800-90Ar1) [#C821] CKG SP 800-133r2 [Vendor affirmed] AES-CBC (SP 800-38A) [#C839] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	Backup session Key	MCO/PCO	E	Success with fips_state = 2 or 3
CN_BACKUP_USERS	Back up the partition users.	AES-CBC (SP 800-38A) [#C839] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	Backup session Key	MCO/PCO	E	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_BACKUP_KEY	Back up the keys.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] AES-CBC (SP 800-38A) [#C839] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	Backup session Key Partition masking key PEK KLK KBK User keys	MCO/PCO	E R R R R R	Success with fips_state = 2 or 3
CN_BACKUP_END	Ends the backup of the partition configuration, keys, and user details.	SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820] RSA SigGen (FIPS 186-4) [#C824]	PAK KBK Backup session Key	MCO/PCO	E G Z	Success with fips_state = 2 or 3
CN_RESTORE_BEGIN	Initiate restoration of partition configuration, users, and keys.	KDF SP 800-108 (KBKDF) [#C826] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	MFKBK OKBK POKBK Backup session Key	MCO/PCO	E E E G	Success with fips_state = 2 or 3
CN_RESTORE_CONFIG	Restore a backed-up partition configuration.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] AES-CBC (SP 800-38A) [#C839] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820] Counter DRBG (SP 800-90Ar1) [#C821] CKG SP 800-133r2 [Vendor affirmed]	Backup session Key	MCO/PCO	E	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_RESTORE_USERS	Restore the partition users.	AES-CBC (SP 800-38A) [#C839] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	Backup session Key	MCO/PCO	E	Success with fips_state = 2 or 3
CN_RESTORE_KEY	Restore the backed-up keys.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	Backup session Key User keys Partition masking key PEK KLK KBK	MCO/PCO	E W W W W W	Success with fips_state = 2 or 3
CN_RESTORE_END	Ends the restoration of backed up partition configuration, keys and user details.	SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	KBK Backup session Key	MCO/PCO	G Z	Success with fips_state = 2 or 3
CN_BACKUP_OBJECT	Backup partition key, partition CSR, PO cert, partition cert signed by PO, user auth keys.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	Backup session Key PAK PAC POAC POTAC CAPubK	MCO/PCO	E R R R R R	Success with fips_state = 2 or 3
CN_WRAP_KBK (Modes: KBK_WRAP_WITH_KEK, KBK_WRAP_WITH_CERT_AUTH_DERIVED_KEY, KBK_WRAP_WITH_RSA)	Wrap KBK out of the HSM.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] KTS-IFC (KTS) (SP 800-56Br2) [#A1194],	KBK, User Keys	PCO	R	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_UNWRAP_KBK (Modes: KBK_WRAP_WITH_KEK, KBK_WRAP_WITH_CERT_AUTH_DERIVED_KEY, KBK_WRAP_WITH_RSA)	Unwraps KBK into the HSM.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] KTS-IFC (KTS) (SP 800-56Br2) [#A1194],	KBK, User Keys	PCO	W	Success with fips_state = 2 or 3
CN_RESTORE_OBJECT	Restore the backed-up object and object details.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	Backup session Key PAK PAC POAC POTAC CAPubK	MCO/PCO	E W W W W W	Success with fips_state = 2 or 3
CN_SET_M_VALUE	Set the current M value for a CO service.	None	None	PCO	None	Success with fips_state = 2 or 3
CN_SET_NODEID	Sets the cluster node ID for a partition.	None	None	PCO/ AU	None	Success with fips_state = 2 or 3
CN_SET_POLICY	Set an HSM policy.	None	None	MCO/PCO	None	Success with fips_state = 2 or 3
CN_CREATE_USER, CN_CREATE_PRE_OFFICER, CN_CREATE_CO, CN_CREATE_APPLIANCE_USER	Create a new CU, CO, Pre-CO or AU user with the provided name and password.	AES [#C819], allowed per IG D.G, AES-CBC (SP 800-38A) [#C819] PBKDF (SP 800-132) [#A1196] RSA SigVer (FIPS 186-4) [#C824]	PMEK 2FAMofNPubK PEK	MCO/PCO	E W E	Success with fips_state = 2 or 3
CN_DELETE_USER	Delete the user with the given name.	None	User Keys	PCO/MCO	Z	Success with fips_state = 2 or 3
CN_LIST_USERS	List all users of the current partition.	None	None	PCO/PCU/AU/UN-AUTH	None	Success with fips_state = 2 or 3
CN_GET_USER_INFO	Get user info and user attributes of a user.	None	None	PCO/ PCU	None	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_UNLOCK_USER	Unlock CU or AU user which got locked up due to invalid login attempts.	None	None	PCO/MCO	None	Success with fips_state = 2 or 3
CN_ALWAYS_AUTHORIZE_USER	Context specific explicit user authorization service for CKA_ALWAYS_AUTHENTICATE keys.	AES-CBC (SP 800-38A) [#C839] RSA SigVer (FIPS 186-4) [#C824] PBKDF (SP 800-132) [#A1196]	PEK CAPubK Login passwords 2FAMofNPubK	PCU	E E E E	Success with fips_state = 2 or 3
CN_CERT_AUTH_GET_SOURCE_RANDOM	Gets the source random number required for mutual trust protocol.	Counter DRBG (SP 800-90Ar1) [#C821]	DRBG ENTROPY/CTR_DRBG Internal State	PCO	R E	Success with fips_state = 2 or 3
CN_CERT_AUTH_VALIDATE_PEER_CERTS	Validates the peer certificates as part of the mutual trust protocol.	Counter DRBG (SP 800-90Ar1) [#C821] RSA SigVer (FIPS 186-4) [#C824]	DRBG ENTROPY/CTR_DRBG Internal State MARC FMAC PAC AOTAC AOAC POTAC POAC	PCO	E	Success with fips_state = 2 or 3
CN_CERT_AUTH_SOURCE_KEY_EXCHANGE	Generate source key exchange message from the HSM.	Counter DRBG (SP 800-90Ar1) [#C821] RSA SigVer (FIPS 186-4) [#C824] KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	SAZ PAK DRBG ENTROPY/CTR_DRBG Internal State	PCO	G R E	Success with fips_state = 2 or 3
CN_CLONE_SOURCE_INIT	Fetch the value for the clone target initialization.	Counter DRBG (SP 800-90Ar1) [#C821] RSA KeyGen (FIPS 186-4) [#C824] ECDSA KeyGen (FIPS 186-4) [#C825]	PCPK DRBG ENTROPY/CTR_DRBG Internal State Partition Cloning Initiator Public Key	PCO	G E G	Success with fips_state = 2 or 3
CN_CLONE_SOURCE_STAGE1	Push clone target output into clone source.	KAS-ECC (KAS) Sp800-56Ar3 [#A1219]	CSSZ PCSK PCSMK Partition masking key	PCO	G G G R E	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
		HMAC-SHA-1 (FIPS 198-1) [#C822], HMAC-SHA2-224 (FIPS 198-1) [#C822], HMAC-SHA2-256 (FIPS 198-1) [#C822], HMAC-SHA2-384 (FIPS 198-1) [#C822], HMAC-SHA2-512 (FIPS 198-1) [#C822] AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	Partition Cloning Responder Public Key			
CN_CLONE_TARGET_INIT	Push clone source output into clone target.	Counter DRBG (SP 800-90Ar1) [#C821] RSA KeyGen (FIPS 186-4) [#C824] ECDSA KeyGen (FIPS 186-4) [#C825]	PCPK DRBG ENTROPY/CTR_DRBG Internal State Partition Cloning Responder Public Key Partition Cloning Initiator Public Key	PCO	G E G E	Success with fips_state = 2 or 3
CN_CLONE_TARGET_STAGE1	Fetch the value for clone target end.	KAS-ECC (KAS) Sp800-56Ar3 [#A1219] HMAC-SHA-1 (FIPS 198-1) [#C822], HMAC-SHA2-224 (FIPS 198-1) [#C822], HMAC-SHA2-256 (FIPS 198-1) [#C822], HMAC-SHA2-384 (FIPS 198-1) [#C822], HMAC-SHA2-512 (FIPS 198-1) [#C822] AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	CSSZ PCSK PCSMK Partition Masking Key	PCO	G G G W	Success with fips_state = 2 or 3
CN_CERT_AUTH_TARGET_KEY_EXCHANGE	Validate key exchange message from peer. Used in cert-based cloning.	KAS-IFC HKDF (SP 800-56Br2) KAS-IFC OneStep (SP 800-56Br2) KAS-IFC TwoStep (SP 800-56Br2) KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	PAK SAZ	PCO	E G	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
		KDF SP 800-108 (KBKDF) [#C826]				
CN_CREATE_OBJECT	Import a public key into the HSM.	None	User Public Keys	PCU	W	Success with fips_state = 2 or 3
CN_GEN_KEY_ENC_KEY	Generates KLK. Generate the key encryption key The type of key is determined by the kek_method parameter in the hsm_config file. The KLK is always a global key.	ECDSA KeyGen (FIPS 186-4) [#C825] KAS-IFC HKDF (SP 800-56Br2) KAS-IFC OneStep (SP 800-56Br2) KAS-IFC TwoStep (SP 800-56Br2) KAS-ECC (KAS) Sp800-56Ar3 [#A1219]	KLK Partition KeyLoading Private Key KLSZ	MCO/PCO	G,W G, E G	Success with fips_state = 2 or 3
CN_EXTRACT_MASKED_OBJECT	Extracts a masked object. i.e. retrieves an object by wrapping it with a masking key shared by the process of cloning.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	User Keys, PEK, KLK, Partition Masking Key	PCU/ PCO/AU CO	R E	Success with fips_state = 2 or 3
CN_INSERT_MASKED_OBJECT	Inserts a masked object into an HSM which Is extracted from another HSM.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] TRIPLE-DES SP 800-38B [#A1198], NO SECURITY CLAIMED PER IG 2.4.A AES-CMAC (SP 800-38B) [#A1195]	User Keys, PEK, KLK, Partition Masking Key	PCU/ PCO/AU	W E	Success with fips_state = 2 or 3
CN_DESTROY_OBJECT	Destroys Key Object.	None	User Keys	PCU	Z	Success with fips_state = 2 or 3
CN_GET_ATTRIBUTE_VALUE	Retrieve single key attribute/metadata.	None	User Keys	PCU	R	Success with fips_state = 2 or 3
CN_GET_ATTRIBUTE_SIZE	Retrieves the size of an attribute of an object.	None	User Keys	PCU	R	Success with fips_state = 2 or 3
CN_GET_ALL_ATTRIBUTE_SIZE	Retrieves the size of all attributes of an object.	None	User Keys	PCU	R	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_GET_ALL_ATTRIBUTE_VALUE	Retrieves all attributes/metadata of an object.	None	User Keys	PCO/PCU	R	Success with fips_state = 2 or 3
CN_MODIFY_OBJECT	Use the setAttribute command to modify object attributes.	None	User Keys	PCO/PCU	W	Success with fips_state = 2 or 3
CN_FIND_OBJECTS_USING_COUNT/CN_FIND_ALL_OBJECTS_IN_RANGE/CN_FIND_ALL_OBJECTS/CN_FIND_ALL_OBJECTS_USING_COUNT/CN_FIND_OBJECTS/CN_FIND_OBJECTS_FROM_INDEX	Finds all key(s) in the partition based on input criteria. An array of key handles will be returned for the keys that match the input criteria specified, key class, key label, etc. Search can be requested from an index.	None	User Keys	MCO/PCO/PCU/AU	R	Success with fips_state = 2 or 3
CN_GENERATE_KEY	Generates a symmetric key of given key type and length.	Counter DRBG (SP 800-90Ar1) [#C821] CKG SP 800-133r2 [Vendor affirmed] TRIPLE-DES SP 800-38B [#A1198], NO SECURITY CLAIMED PER IG 2.4.A AES-CMAC (SP 800-38B) [#A1195]	Symmetric User Keys DRBG Entropy/CTR_DRBG Internal State	PCU	G E	Success with fips_state = 2 or 3
CN_SPLIT_SECRET_KEY	Split a symmetric key into multiple keys based on the attributes given by the user. The resulting key handles are stored as output in splitKeyArgs structure.	None	Symmetric User Keys	PCU	G	Success with fips_state = 2 or 3
CN_GENERATE_KEY_PAIR	Generate asymmetric keys (RSA/DSA/ECC). Updates the public and private key handles in the output on return.	RSA KeyGen (FIPS 186-4) [#A1199], RSA KeyGen (FIPS 186-4) [#C824] ECDSA KeyGen (FIPS 186-4) [#C825] ECDSA KeyVer (FIPS 186-4) [#C825] DSA KeyGen (FIPS 186-4) [#C823] DSA PQGGen (FIPS 186-4) [#C823] DSA PQGVer (FIPS 186-4) [#C823]	Asymmetric User Keys	PCU	G	Success with fips_state = 2 or 3
CN_EXPORT_PUB_KEY	Export a public key in PEM-encoded format.	None	User Keys	PCU	R	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_SHARE_OBJECT	Share an object between users.	None	User Keys	PCU	W	Success with fips_state = 2 or 3
CN_GET_OBJECT_INFO	Obtains Key details like shared sessions, shared users and m_values of USE_KEY, MANAGE_KEY services.	None	User Keys	PCU	R	Success with fips_state = 2 or 3
CN_TOMBSTONE_OBJECT	Marks the specified object stored on the HSM invalid.	None	User Keys	PCU	W	Success with fips_state = 2 or 3
CN_DELETE_TOMBSTONED_OBJECT	Used to delete the Tombstone object, Regular Delete will fail if the object is tomb stoned.	None	User Keys	PCU	Z	Success with fips_state = 2 or 3
CN_UNWRAP_KEY/CN_UNWRAP_KEY2	Unwraps a key with an AES/Triple-DES/RSA-Private key existing on HSM or KLK. Takes the output wrapped data of wrapKey2 command.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] TDES-KW (SP 800-38F) [#C1263] Triple-DES-KW (KTS) (SP 800-38F) [#C1263] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] KTS-IFC (KTS) (SP 800-56Br2) [#A1194] KDA HKDF Sp800-56Cr1 [#A1192] KDA OneStep Sp800-56Cr1 [#A1192] KDA TwoStep Sp800-56Cr1 [#A1192] AES-GCM (KTS) (SP 800-38D) [#C839] AES [#C819], allowed per IG D.G AES [#C819], allowed per IG D.G ECDSA KeyVer (FIPS 186-4) [#C825] DSA PQGVer (FIPS 186-4) [#C823] TRIPLE-DES SP 800-38B [#A1198], NO SECURITY CLAIMED PER IG 2.4.A AES-CMAC (SP800-38B) [#A1195]	User Keys, KLK	PCU	E	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
		AES-GCM (KTS) (SP 800-38D) [#A1203] KAS KDA HKDF (SP 800-56Ar3) KAS KDA ONESTEP (SP 800-56Ar3) KAS KDA TWOSTEP (SP 800-56Ar3) KAS ANS 9.63 (SP 800-56Ar3) KAS-KDF-HKDF (SP 800-56Ar3) KAS-KDF-OneStep(SP 800-56Ar3) KAS-KDF-TwoStep (SP 800-56Ar3) KAS-KDF-ANS9.63 (SP 800-56Ar3)				
CN_WRAP_KEY/CN_WRAP_KEY2	Wrap sensitive (private and symmetric) keys from the HSM to the host.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] KDA HKDF Sp800-56Cr1 [#A1192] KDA OneStep Sp800-56Cr1 [#A1192] KDA TwoStep Sp800-56Cr1 [#A1192] AES-GCM (KTS) (SP 800-38D) [#A1203] Counter DRBG (SP 800-90Ar1) [#C821] CKG SP 800-133r2 [Vendor affirmed] AES-GCM (KTS) (SP 800-38D) [#C839] KTS-IFC (KTS) (SP 800-56Br2) [#A1194] KAS KDA HKDF (SP 800-56Ar3) KAS KDA ONESTEP (SP 800-56Ar3) KAS KDA TWOSTEP (SP 800-56Ar3) KAS ANS 9.63 (SP 800-56Ar3) KAS-KDF-HKDF (SP 800-56Ar3)	User Keys, KLK	PCU	E	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
		KAS-KDF-OneStep(SP 800-56Ar3) KAS-KDF-TwoStep (SP 800-56Ar3)KAS-KDF-ANS9.63 (SP 800-56Ar3) KAS-KDF-ANS9.63 (SP 800-56Ar3)				
CN_NIST_AES_WRAP_UNWRAP/ CN_NIST_AES_WRAP_UNWRAP2	Wrap/unwrap data with a specified AES key.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	Symmetric User Keys	PCU	E	Success with fips_state = 2 or 3
CN_GET_RSA_CACHE_SIZE	Get the number of available RSA keys.	N/A	Asymmetric User Keys	MCO	N/A	Success with fips_state = 2 or 3
CN_DERIVE_KEY	Derives a key using a supported KDF mechanism with the params given from the user.	KDF SP 800-108 (KBKDF) [#C826], KDF SP 800-108 (KBKDF) [#C839], KDF SP 800-108 (KBKDF) [#A1191] KDF ANS 9.63 (CVL) (SP 800-135r1) [#C825] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] KDA HKDF Sp800-56Cr1 [#A1192] KDA OneStep Sp800-56Cr1 [#A1192] KDA TwoStep Sp800-56Cr1 [#A1192] KAS-ECC-SSC Sp800-56Ar3 [#A1220] KAS KDA HKDF (SP 800-56Ar3) KAS KDA ONESTEP (SP 800-56Ar3) KAS KDA TWOSTEP (SP 800-56Ar3) KAS ANS 9.63 (SP 800-56Ar3)	User Keys	PCU	G E	Success with fips_state = 2 or 3
CN_MODIFY_KEY_OWNER	Modify the owner user of a key.	None	User keys	PCO	W	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_ADMIN_GET_PARTN_KEYHANDLES_HASH	Gets Hash of all keys for a partition.	SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	User Keys	MCO/MCU/ PCU/AU	R E	Success with fips_state = 2 or 3
CN_GET_PARTN_SINGLE_KEYHANDLE_HASH	Gets Hash of single key for a partition.	SHA-1 (FIPS 180-4) [#C820], SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	User Keys	PCO/AU	R	Success with fips_state = 2 or 3
CN_PARK_OBJECT	Park a key using the given parking key. Only parkable keys can be parked. Keys with parkable attribute not set cannot be parked.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	User Keys	PCU	R E	Success with fips_state = 2 or 3
CN_UNPARK_OBJECT	Unpark given parked object using the given parking key.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	User Keys	PCU	W E	Success with fips_state = 2 or 3
CN_SET_USER_ATTR	Set User attributes which control the functionality of a crypto user.	None	None	PCO	None	Success with fips_state = 2 or 3
CN_LIST_AUTH_PUB_KEYS	List all registered user auth pub keys.	None	None	PCO	None	Success with fips_state = 2 or 3
CN_CERT_AUTH_REMOVE_CERT	Stores or removes the Partition TA cert.	None	None	PCO	W Z	Success with fips_state = 2 or 3
CN_PARTN_GET_AUDIT_DETAILS	Gets Audit Logs Details.	None	None	AU/PCO	None	Success with fips_state = 2 or 3
CN_PARTN_GET_AUDIT_LOGS	Gets Audit Logs.	SHA2-256 (FIPS 180-4) [SHS #1780]	None	AU/PCO	None	Success with fips_state = 2 or 3
CN_PARTN_GET_AUDIT_SIGN	Gets Audit Logs Hash or RSA signature.	SHA2-256 (FIPS 180-4) [SHS #1780] RSA Signature Primitive (CVL) (FIPS 186-4) [#C839]	PAK	AU/PCO	E (PAK)	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_PARTN_ACK_AUDIT_SIGN	Acks previously retrieved signature. Either hash or signature needs to match with the values stored by HSM for firmware to accept the signature acknowledgement.	None	None	AU/PCO	None	Success with fips_state = 2 or 3
CN_FINALIZE_LOGS	Finalize logs by inserting end marker. No more loggable commands are allowed on the partition after this command is run.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_SIGN	Generate a signature on the given data with the specified private key.	RSA SigGen (FIPS 186-4) [#A1199], RSA SigGen (FIPS 186-4) [#C824] ECDSA SigGen (FIPS 186-4) [#C825] ECDSA SigGen (FIPS 186-4) (CVL) [#C825] DSA SigGen (FIPS 186-4) [#C823] SHA2-224 (FIPS 180-4) [#C820], SHA2-256 (FIPS 180-4) [#C820], SHA2-384 (FIPS 180-4) [#C820], SHA2-512 (FIPS 180-4) [#C820]	Asymmetric User Keys	PCU	E	Success with fips_state = 2 or 3
CN_VERIFY	Verify the signature on the given data with specified public key.	RSA SigVer (FIPS 186-4) [#A1199], RSA SigVer (FIPS 186-4) [#C824] ECDSA SigVer (FIPS 186-4) [#C825] DSA SigVer (FIPS 186-4) [#C823]	User Public Keys	PCU	E	Success with fips_state = 2 or 3
CN_ECC_DH	Computes the shared secret (Z).	KAS-ECC-SSC Sp800-56Ar3 [#A1220]	Asymmetric User Keys	PCU	E	Success with fips_state = 2 or 3
CN_NIST_AES_WRAP	Wrap data with a specified AES key.	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	KLK Symmetric User Keys	PCU	E R	Success with fips_state = 2 or 3
CN_ALLOC_SSL_CTX	Allocates a context segment in the HSM memory and returns a reference to the application for the same.	None	None	PCU	None	Success with fips_state = 2 or 3
CN_FREE_SSL_CTX	Free a context segment for use by another SSL connection.	None	None	PCU	None	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_GEN_PMK	Generate random premaster secret data and writes it into given ctx pointer.	Counter DRBG (SP 800-90Ar1) [#C821]	DRBG ENTROPY/CTR_DRBG Internal State	PCU	E	Success with fips_state = 2 or 3
CN_FIPS_RAND	Generates FIPS random number of given length.	Hash DRBG (SP 800-90Ar1) [#C830]	DRBG Entropy/HASH_DRBG Internal State	PCU	E	Success with fips_state = 2 or 3
CN_ME_PKCS_LARGE	ModExp and PKCS#1, v2.2 encryption and decryption.	RSA Decryption Primitive (CVL) (SP 800-56Br2) [#C839] RSA Signature Primitive (CVL) (FIPS 186-4) [#C839] KTS-IFC (KTS) (SP 800-56Br2) [#A1194] RSA Decryption Primitive (CVL) (SP 800-56Br2)[#A1200]	Asymmetric User Keys	PCU	E	Success with fips_state = 2 or 3
CN_ME_PKCS	ModExp and PKCS#1, v2.2 encryption, decryption, sign and verify.	RSA Decryption Primitive (CVL) (SP 800-56Br2) [#C839] RSA Signature Primitive (CVL) (FIPS 186-4) [#C839] KTS-IFC (KTS) (SP 800-56Br2) [#A1194] RSA Decryption Primitive (CVL) (SP 800-56Br2)[#A1200]	Asymmetric User Keys	PCU	E	Success with fips_state = 2 or 3
CN_FECC	ECDSA Sign/verify and Point add/double/mul operation.	ECDSA SigVer (FIPS 186-4) [#C829] ECDSA SigGen (FIPS 186-4) (CVL) [#C829]	Asymmetric User Keys	PCU	E	Success with fips_state = 2 or 3
CN_HASH	Computes SHA Hash.	SHA-1 (FIPS 180-4) [SHS #1780], SHA2-224 (FIPS 180-4) [SHS #1780], SHA2-256 (FIPS 180-4) [SHS #1780], SHA2-384 (FIPS 180-4) [SHS #1780], SHA2-512 (FIPS 180-4) [SHS #1780]	None	PCU	None	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_HMAC	Compute/Verify the MAC of a complete message. HMAC max message length supported will vary based on Hash type.	HMAC-SHA-1 (FIPS 198-1) [#C839], HMAC-SHA2-224 (FIPS 198-1) [#C839], HMAC-SHA2-256 (FIPS 198-1) [#C839], HMAC-SHA2-384 (FIPS 198-1) [#C839], HMAC-SHA2-512 (FIPS 198-1) [#C839] AES-CMAC (SP 800-38B) [#C839] AES-CMAC (SP 800-38B) [#A1190]	Symmetric and HMAC User Keys	PCU	E	Success with fips_state = 2 or 3
CN_ENCRYPT_DECRYPT	AES encryption and decryption, Triple-DES decryption.	AES-CBC (SP 800-38A) [#C839], AES-CTR (SP 800-38A) [#C839], AES-ECB (SP 800-38A) [#C839] AES-CCM (SP 800-38C) [#C839] AES-GCM (SP 800-38D) [#A1203] AES-GCM (SP 800-38D) [#C839] AES-GMAC (SP 800-38D) [#C839] TDES-CBC (SP 800-38A) [TDES #1311] *legacy use only TDES-ECB (SP 800-38A) [TDES #1311] *legacy use only	Symmetric User Keys	PCU	E	Success with fips_state = 2 or 3
MAJOR_OP_OTHER	When not (RSA <= 4096), do a full handshake. The pre-master secret is read from the context or input and the rest of the handshake is completed. This is used by both the server and the client.	KDF TLS (CVL) (SP 800-135r1) [#C840] KAS-ECC-SSC Sp800-56Ar3 [#A2161] SHA-1 (FIPS 180-4) [SHS #1780], SHA2-224 (FIPS 180-4) [SHS #1780], SHA2-256 (FIPS 180-4) [SHS #1780], SHA2-384 (FIPS 180-4) [SHS #1780], SHA2-512 (FIPS 180-4) [SHS #1780] AES-GCM (SP 800-38D) [#C839]	User Keys	PCU	E	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
MAJOR_OP_FINISHED	Finish off the handshake hash and generate the finished messages for a full handshake. This is used in a full handshake with client authentication on either the client or the server.	SHA-1 (FIPS 180-4) [SHS #1780], SHA2-224 (FIPS 180-4) [SHS #1780], SHA2-256 (FIPS 180-4) [SHS #1780], SHA2-384 (FIPS 180-4) [SHS #1780], SHA2-512 (FIPS 180-4) [SHS #1780] AES-GCM (SP 800-38D) [#C839]	User Keys	PCU	E	Success with fips_state = 2 or 3
MAJOR_OP_RESUME	Completes a resume on either the client or the server. The handshake message data for this request should include all handshake message data from (and including) the most-recent client hello message up until (but not including) the first finished message.	SHA-1 (FIPS 180-4) [SHS #1780], SHA2-224 (FIPS 180-4) [SHS #1780], SHA2-256 (FIPS 180-4) [SHS #1780], SHA2-384 (FIPS 180-4) [SHS #1780], SHA2-512 (FIPS 180-4) [SHS #1780] AES-GCM (SP 800-38D) [#C839]	User Keys	PCU	E	Success with fips_state = 2 or 3
MAJOR_OP_ENCRYPT_DECRYPT_RECORD	Encrypt/decrypt records. Send encrypted E2E request to the FW.	Hash DRBG (SP 800-90Ar1) [#C830] AES-GCM (SP 800-38D) [#C839]	DRBG Entropy/HASH_DRBG Internal State Symmetric User Keys	PCU	E	Success with fips_state = 2 or 3
CN_SHA3	Computes SHA3 Hash.	SHA3-224 (FIPS 202) [#A1197], SHA3-256 (FIPS 202) [#A1197], SHA3-384 (FIPS 202) [#A1197], SHA3-512 (FIPS 202) [#A1197], SHAKE-128 (FIPS 202) [#A1197], SHAKE-256 (FIPS 202) [#A1197]	None	PCU	None	Success with fips_state = 2 or 3
MAJOR_OP_DECRYPT_AND_ENCRYPT	Performs decryption with one cipher and re-encrypts the decrypted data with another cipher.	AES-CBC (SP 800-38A) [#C839], AES-CTR (SP 800-38A) [#C839], AES-ECB (SP 800-38A) [#C839] AES-CCM (SP 800-38C) [#C839] AES-GCM (SP 800-38D) [#A1203] AES-GCM (SP 800-38D) [#C839] AES-GMAC (SP 800-38D) [#C839] TDES-CBC (SP 800-38A) [TDES #1311] TDES-ECB (SP 800-38A) [TDES #1311]	Symmetric User Keys	PCU	E	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_GET_TOKEN	Gets a token or token info from the partition for a given service.	Counter DRBG (SP 800-90Ar1) [#C821]	DRBG ENTROPY/CTR_DRBG Internal State	PCO/PCU	E	Success with fips_state = 2 or 3
CN_APPROVE_TOKEN	Submit approvals on token, approval could be on a single or multiple blobs.	RSA SigVer (FIPS 186-4) [#C824]	2FAMofNPubK	PCO/PCU	E	Success with fips_state = 2 or 3
CN_LIST_TOKENS	List all MofN tokens in the current partition.	None	None	PCO/PCU	None	Success with fips_state = 2 or 3
CN_TOKEN_TIMEOUT	Get or set the timeout values of the tokens in the partition.	None	None	PCO	None	Success with fips_state = 2 or 3
CN_DELETE_TOKEN	Deletes the existing MxN tokens based on the token-delete options.	None	None	PCO/PCU	None	Success with fips_state = 2 or 3
CN_SM_IMAGE_DELETE	Delete Secure Machine images from the partition.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_SME_DIAG_INFO	To get the SME diagnostic information about SM Linux and SM Manager.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_SET_SM_APP_CONFIG	Set Secure Machine App Configuration. It is used to allocate resources for SMApp.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_GET_SM_APP_CONFIG	To get the resources allocated to an SMApp.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_SET_SM_CAPABILITY	Set Secure Machine Capability It is used to enable or disable the Capability.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_GET_SM_CONFIG	To get the existing configuration and other information. Total SMApp resources used resources information is available.	None	None	MCO	None	Success with fips_state = 2 or 3
CN_SMAPP_DIAG_INFO	To get the SMApp diagnostic information such as CPU, memory, statistics, etc.	None	None	PCO	None	Success with fips_state = 2 or 3

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to keys and/or SSPs	Indicator
CN_SMAPP_UPDATE_BEGIN	Initiates the SM App firmware update on a partition. This requires CO to be logged in.	None	None	PCO	None	Success with fips_state = 2 or 3
CN_SMAPP_UPDATE	Loading/Updating an SMApp on.	None	None	PCO	None	Success with fips_state = 2 or 3
CN_SMAPP_UPDATE_END	Completion of SMApp firmware update on a partition. This requires CO to be logged in.	RSA SigVer (FIPS 186-4) [#C824]	POTAC	PCO	E (POTAC)	Success with fips_state = 2 or 3
CN_SMAPP_CTRL	Does SM app start/stop/status on a partition. Requires PCO to be logged in.	None	None	PCO	None	Success with fips_state = 2 or 3
CN_SMAPP_DELETE	Performs SMApp delete. This requires CO to be logged in.	None	None	PCO	None	Success with fips_state = 2 or 3
CN_SMAPP_WRITE_DATA	Store SMApp data into SMApp private storage.	None	None	PCU	None	Success with fips_state = 2 or 3
CN_SMAPP_READ_DATA	Read SMApp data from SMApp private storage.	None	None	PCU	None	Success with fips_state = 2 or 3
CN_SMAPP_DELETE_FILE	Read SMApp data from SMApp private storage.	None	None	PCU	None	Success with fips_state = 2 or 3
CN_SET_SM_CONFIG	To configure resources for the SM..	None	None	MCO	None	Success with fips_state = 2 or 3
CN_SMAPP_WRITE_SFRAM	Write SMApp data into SMApp private SFRAM memory.	None	None	PCU	None	Success with fips_state = 2 or 3
CN_SMAPP_READ_SFRAM	Read SMApp data from SMApp private SFRAM memory.	None	None	PCU	None	Success with fips_state = 2 or 3
CN_LIST_UNLINKED_OBJECTS	Return the total tombstone sessions, keys and contexts.	None	None	PCO	None	Success with fips_state = 2 or 3

Indicator for Approved Services:

All Approved services can be executed in the Non-Approved services mode with indicator of “success with fips_state=0”. Approved services that also support the use of non-approved security functions are enumerated in the Non-Approved Services table below with their supported non-approved security functions.

The indicator is success for all approved services when the partition is operated in the Approved mode.

Table 14 – Non-Approved Services

Service	Description	Algorithms Accessed	Role	Indicator
CN_GENERATE_PBE_KEY	Generate PBE Triple-DES key with the given password, salt, and iteration count. Make sure that HSM is initialized with fips_state=0. The fips_state parameter can be found in the hsm_config file.	COUNTER DRBG Allowed Per IG 2.4.A/ PBE	PCU	SUCCESS with fips_state = 0
LSPAY_GENERATE_ASYMM_KEY	(Generates RSA KEY Pair (mod_len>= 2048bit)) Generates EC KEY PAIR (Curves: Nist P256, 224, 384, 521, Brain pool, x25519/448 and Decp256K1 and FRP256v1)	RSA (non-compliant) ECDSA (non-compliant)/ KAS-ECC (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_GENERATE_SYMM_KEY	Generates symmetric key AES/TDEA Keys used for LSPay operations.	COUNTER DRBG Allowed Per IG 2.4.A	PCU	SUCCESS with fips_state = 0
LSPAY_EXPORT_PUBLIC_KEY	Exports Public key for RSA BYOK.	N/A	PCU	SUCCESS with fips_state = 0
LSPAY_IMPORT_PUBLIC_KEY	Imports RSA public Key for RSA BYOK	N/A	PCU	SUCCESS with fips_state = 0
LSPAY_VALIDATE_PUBLIC_KEY	Validates RSA public Key.	RSA (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_IMPORT_KPK	Imports OAEP wrapped or ECDH_AES_PAD wrapped symmetric key.	RSA (non-compliant), EC-AES / ECDH KDF	PCU	SUCCESS with fips_state = 0
LSPAY_EXPORT_KPK	Exports OAEP wrapped symmetric Key from HSM.	RSA (non-compliant), EC-AES/ ECDH KDF	PCU	SUCCESS with fips_state = 0
LSPAY_IMPORT_KEY	Import symmetric or asymmetric keys. Wrap mech: TR31, AES_CBC, AES_CBC_PAD.	AES (non-compliant) Triple-DES (non-compliant).	PCU	SUCCESS with fips_state = 0
LSPAY_IMPORT_TR34_KEY	Import symmetric keys using TR-34 unwrap.	RSA (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_EXPORT_KEY	Exports Symmetric key Wrapped with TR31/AES_CBC/AES_CBC_PAD.	AES (non-compliant) Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_EXPORT_TR34_KEY	Exports symmetric keys wrapped with TR34 mechanism.	RSA (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_TRANSLATE_KEY	Translates wrapped Key from on KPK to other KPK.	AES (non-compliant) Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_IMPORT_CERT	Imports peer’s certificate to read public key required in TR34. Import X901 Certificate into HSM.	N/A	PCU	SUCCESS with fips_state = 0
LSPAY_IMPORT_DECIMAL_TABLE	Imports encrypted decimal table to be used in PIN APIs to decimalize native PIN.	AES (non-compliant)/Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0

Service	Description	Algorithms Accessed	Role	Indicator
LSPAY_GENERATE_CSR	CREATE CSR with given Key Pair.	RSA (non-compliant) /ECDSA (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_DERIVE_KEY	Derives DUKPT working key from the BDK.	AES (non-compliant)/DES/ Triple-DES (non-compliant))	PCU	SUCCESS with fips_state = 0
LSPAY_ENCRYPT	Encrypts input data or PIN.	AES (non-compliant)/Triple-DES (non-compliant) /DES/ Double-DES	PCU	SUCCESS with fips_state = 0
LSPAY_DECRYPT	Decrypts input data or PIN.	AES (non-compliant)/Triple-DES (non-compliant) /DES/ Double-DES	PCU	SUCCESS with fips_state = 0
LSPAY_DECRYPT_THEN_ENCRYPT	Decrypts the input cipher text with one key and encrypts with another key.	AES (non-compliant)/Triple-DES (non-compliant) /DES/ Double-DES	PCU	SUCCESS with fips_state = 0
LSPAY_MAC_GEN	Computes MAC on input data. Algorithm used: DES/Triple-DES	DES MAC / Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_MAC_VERIFY	Verifies MAC with calculated AMC on input data.	DES MAC / Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_MAC_TRANSLATE	Translates MAC by using new Key on input data.	DES MAC / Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_FPE_ENCRYPT	Performs FPE FF1/FF3-1 Encrypt operation on input data.	AES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_FPE_DECRYPT	Performs FPE FF1/FF3-1 Decrypt operation on input data.	AES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_SIGN	Performs Sign and Verify on input data.	RSA (non-compliant), EDDSA (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_SIGN_VERIFY	Verifies sign on input data.	RSA (non-compliant), EDDSA (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_PIN_BLOCK_TRANSLATE	Decrypts the input PIN using decryption key, translates to given PIN format and encrypts with another key.	AES (non-compliant)/Triple-DES (non-compliant) RSA (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_DERIVE_PIN_FROM_OFFSET	Derive PIN from given offset. Encrypts validation data with DES EDE. Derives native PIN, then offset will be added to derive IBM PIN. PIN will be encoded in given ISO format. Encrypt encoded PIN with PIN encryption key.	AES (non-compliant)/Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_DERIVE_OFFSET_FROM_PIN	Generates IBM offset from given PIN. Decrypt and decode received PIN. Generate native from given validation data. Subtract decoded PIN from native PIN t to get PIN offset.	AES (non-compliant)/Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0

Service	Description	Algorithms Accessed	Role	Indicator
LSPAY_VERIFY_PIN	Verifies given PIN. Decrypt and decode received PIN. Generate native from given validation data. Add offset to native PIN. Compare resultant PIN with received PIN.	AES (non-compliant)/Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_PVV_GENERATION	Perform PVV generation on PIN and PAN data.	AES (non-compliant)/Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_PVV_VERIFY	Verifies given PVV.	AES (non-compliant)/ Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_EMV_GENVERIFY_AC	Perform EMV crypto operations. Generate ARPC. Generate or Verify ARQC.	AES (non-compliant) Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_EMV_SECURE_MSG_GEN	Generates MAC over secure message.	AES (non-compliant) Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_CVV_GEN	Generates CVV, CVV2, iCVV on given card details	AES (non-compliant)/Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_CVV_VERIFY	Verifies CVV with given card details.	Triple-DES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_KEY_SHARE_CREATE	Creates components of key.	Shamir's Key share	PCU	SUCCESS with fips_state = 0
LSPAY_KEY_SHARE_EXPORT_KEY_COMPONENT	Exports created components in encrypted format.	AES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_KEY_SHARE_COMBINE_INIT	Starts combine key init.	N/A	PCU	SUCCESS with fips_state = 0
LSPAY_KEY_SHARE_IMPORT_COMPONENT	Import component of the key.	AES (non-compliant)	PCU	SUCCESS with fips_state = 0
LSPAY_KEY_SHARE_COMBINE_KEY	Combines all components of the key.	Shamir's key share	PCU	SUCCESS with fips_state = 0
LSPAY_KEY_SHARE_ZEROIZE	Erases all components of the key.	N/A	PCU	SUCCESS with fips_state = 0
LSPAY_MFK_GENERATE	Generates MFK key.	COUNTER DRBG Allowed Per IG 2.4.A	PCO	SUCCESS with fips_state = 0
LSPAY_MFK_GET_INFO	Returns MFK information for partition.	N/A	PCO	SUCCESS with fips_state = 0
LSPAY_MFK_DELETE	Deletes MFK.	N/A	PCO	SUCCESS with fips_state = 0
LSPAY_MFK_SET_PRIMARY	Set MFK as primary.	N/A	PCO	SUCCESS with fips_state = 0
MAJOR_OP_RSASERVER_LARGE	Does a full handshake on the server with RSA > 1024 and <= 4096. This is used in a full handshake on the server.	RSA (non-compliant)	PCU	SUCCESS with fips_state = 0
MAJOR_OP_RSASERVER	Does a full handshake on the server with RSA >=512 and <= 1024. This is used in a full handshake on the server.	RSA (non-compliant)	PCU	SUCCESS with fips_state = 0
Firmware Update	Vendor zeroizes the HSM and updates the bootloader.	N/A	UN-AUTH	RED LED BLINK

Service	Description	Algorithms Accessed	Role	Indicator
CN_GENERATE_KEY_PAIR (non-compliant)	Generates asymmetric keys (RSA/ ECC). Updates the public and private key handles in the output on return. Caveats in Non Approved apart from approved mode: RSA 1024 bits allowed along with all odd public exponent; i.e., even lesser than 65537. • NID_X9_62_prime192v 1/NID_sect163k1/NID_ED25519/ • NID_sect163r2 /NID_secp192k1/NID_brainpoolP160r1/ • NID_brainpoolP192r1 /NID_X25519/ • NID_X448	RSA (non-compliant) ECDSA (non-compliant) KAS-ECC (non-compliant)	PCU	Success with fips_state = 0
CN_GENERATE_KEY (non-compliant)	Generates a symmetric key of given key type and length. Caveats in Non-Approved apart from approved mode: DES token key is allowed	COUNTER DRBG Allowed Per IG 2.4.A	PCU	Success with fips_state = 0
CN_CREATE_OBJECT (non-compliant)	Imports a public key into HSM. Caveats in Non Approved apart from approved mode: • RSA 1024 bits allowed • NID_ED25519/ NID_secp192k1/ NID_brainpoolP160r1/ • NID_brainpoolP192r1/ NID_X25519/NID_X448	None	PCU	Success with fips_state = 0
CN_UNWRAP_KEY (non-compliant)	Unwraps a key with an AES/ Triple-DES/RSA private key existing on HSM or KLK. Takes the output wrapped data of wrapKey2 command. Caveats in Non Approved apart from approved mode: • RSA 1024 bit • RSA PKCS1V1.5 Unwrap • NID_X9_62_prime192v1/ NID_sect163k1/ NID_ED25519/ • NID_sect163r2 / NID_secp192k1/ NID_brainpoolP160r1/ • NID_brainpoolP192r1 / NID_X25519/ • NID_X448 • Triple-DES	RSA (non-compliant), EC-AES / ECDH KDF AES (non-compliant) Triple-DES (non-compliant).	PCU	Success with fips_state = 0
CN_WRAP_KEY (non-compliant)	Wraps sensitive (private and symmetric) keys from the HSM to the host. Caveats in Non Approved apart from approved mode: • AES-ECB mode • AES-CBC mode • AES-CBC-PAD mode • Triple-DES ECB mode • Triple-DES CBC mode	AES (non-compliant) Triple-DES (non-compliant) RSA (non-compliant)	PCU	Success with fips_state = 0

Service	Description	Algorithms Accessed	Role	Indicator
	• Triple-DES NIST Wrap mode • RSA-PKCS1V1.5 Wrap			
CN_EXTRACT_MASKED_OBJECT (non-compliant)	Extracts a masked object; i.e., retrieves an object by wrapping it with a masking key shared by the process of cloning.	AES (non-compliant)	PCU PCO AU	Success with fips_state = 0
CN_STORE_FW_SIGNING_KEY (non-compliant)	Configure an RSA or EC public key into HSM as AO attestation key. These keys can be of modulus 1024, 2048, 3072, and 4096 or a supported 256 bits, 384 bits or 521 bits EC curve . Caveats in Non Approved is 192 bit curves supported	RSA (non-compliant) ECDSA (non-compliant)	MCO	Success with fips_state = 0
CN_ME_PKCS_LARGE (non-compliant) CN_ME_PKCS (non-compliant)	ModExp and PKCS#1v1.5 and PKCS#1v2.2 Sign and verify. PKCS#1v1.5 and PKCS#1v2.2 encrypt and decrypt	RSA (non-compliant)	PCU	Success with fips_state = 0
CN_STORE_VENDOR_PRE_SHARED_KEY (CN_STORE_KBK_SHARE) (non-compliant)	Stores fixed keys (KBK) for backup. Including PKCS#1v1.5	RSA (non-compliant	Manufacturer	Success with fips_state = 0
CN_INSERT_MASKED_OBJECT (non-compliant)	Inserts a masked object into an HSM that Is extracted from another HSM.	AES (non-compliant) Triple-DES (non-compliant)	PCU PCO AU	Success with fips_state = 0
CN_ENCRYPT_SESSION (non-compliant)	Enables encrypted communication channel. Caveat is Non Approved mode allow the additional Cipher suite E2E_RSA_AES128_GCM_SHA256 and E2E_RSA_AES128_GCM_SHA384	RSA (non-compliant), EC-AES / ECDH KDF	UN-AUTH	Success with fips_state = 0
CN_DERIVE_KEY (non-compliant)	Derives a key using a supported KDF mechanism with the params given by the user.	AES (non-compliant) Triple-DES (non-compliant) RSA (non-compliant) EC-AES / ECDH KDF	PCU	Success with fips_state = 0

Indicator for Non-Approved Services:

The indicator is success for all Non-Approved services only when the partition is operated in the non-approved mode. The Non-Approved services will fail with RET_POLICY_MISMATCH when the partition is operated in the approved mode.

5 Software/Firmware Security

During the bootup, the following integrity tests are run:

1. Bootloader runs a 32-bit CRC verification algorithm to validate the bootloader image itself.
2. Then, bootloader runs the firmware Integrity tests based on RSA 2048-bit SHA2-256 signature.
3. Integrity tests are part of POST and can be triggered by MCO through the CN_INVOKE_FIPS service.
4. Module Firmware has two components, Firmware (FW) and Secure Machine World (SMW) as listed in Section 2.1. The two components are as follows:

   a. Firmware (FW): CNN35XX-NFBE-FW-2.09-0702
   b. Secure Machine World : CNN35XX-NFBE-SMW-2.09-0702

5. Module Firmware is binary executable and is not open source.

6 Operational Environment

The module implements a limited operational environment. Area 6 modifiable Operational Environment requirements do not apply to the module in this validation.

The module runs SMP Linux 4.9 which is part of the firmware image CNN35XX-NFBE-FW-2.09-0702.

1. The image does not run arbitrary applications or create new application flows. All the execution flows are pre-defined for each service in the module.
2. The image runs a monolithic application and manages the cryptographic software solely through intended services.

7 Physical Security

7.1 Physical Security Mechanisms

The module's cryptographic boundary is defined to be the outer perimeter of the hard epoxy enclosure containing the hardware and firmware components. The module is opaque and completely conceals the internal components of the cryptographic module. The epoxy enclosure of the module prevents physical access to any of the internal components without having to destroy the module. There are no operator-required actions.

7.2 Tamper Evidence

The module is coated in hard epoxy, such that any physical breach attempt leaves behind evidence of tamper. This is shown in the figure below.

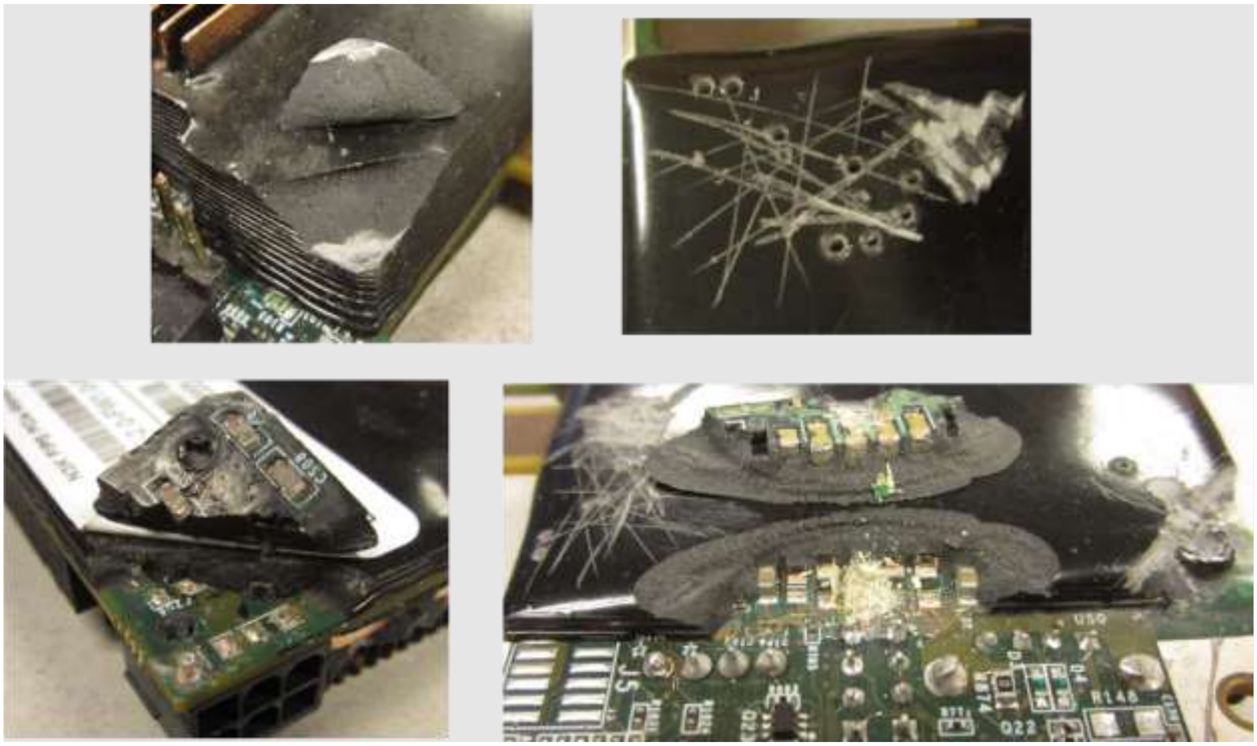


Figure 4 – Cryptographic Module Showing Tamper Evidence

Top: Minor tamper to the epoxy only

Bottom: Major tamper, damaging circuitry

While the module is designed to prevent successful tampering (any physical breach to module circuitry is likely to destroy the module, as per FIPS 140-3 Level 3 Physical Security requirements), the module should still be checked periodically for attempts. Guidelines are provided in the table below.

Table 15 – Physical Security Inspection Guidelines

Physical Security Mechanism	Recommended Frequency of Inspection/Test	Inspection/Test Guidance Details
Epoxy Coating	12 Months	Examine surface of module for scratched or damaged epoxy, especially if circuitry shows.

If the module is found to be meaningfully damaged or tampered with (e.g., circuitry is showing, or other significant damage has occurred), it should be removed from usage and destroyed.

Table 16 – EFP/EFT

	Temperature or voltage measurement	Specify EFP or EFT	Specify if this condition results in a shutdown or zeroization
Low Temperature	-5C	EFP	Shutdown
High Temperature	90C	EFP	Shutdown
Low Voltage	2.9V	EFT	Shutdown
High Voltage	18.8V	EFT	Shutdown

Table 17 – Hardness Testing Temperature ranges

	Hardness tested temperature measurement
Low Temperature	-35C
High Temperature	100C

8 Non-Invasive Security

At the time of this validation, no approved non-invasive attack mitigation test metrics are defined.

9 Sensitive Security Parameters Management

9.1 Definition of Critical Security Parameters (CSPs)

The Manufacturer FIPS Data Encryption Key (MFDEK) and HSM Master Partition Master Encryption Key are stored in plaintext form in the EEPROM. The Partition Master Encryption Key (PMEK) is stored encrypted under the HSM Master Partition Master Encryption Key. All other keys and CSPs stored in the persistent memory are encrypted by the MFDEK, HSM Master Partition Master Encryption Key, or PMEK. All general-purpose user CSPs are generated/created by the PCU and these CSPs can be shared between multiple PCUs.

The module itself enforces that the SSPs cannot be shared between the approved and non-approved modes.

Note:

- The SSPs are zeroized securely by writing zeros to memory when in zeroization.
- Private or secret keys are always encrypted when doing export /import, and key encapsulation mechanisms are listed in the respective CSP row of Table 18 – SSPs.
- All CSPs from Table 18 are entered through automated electronic entry mechanisms.

The below notations are used to indicate the CSP zeroization and memory de-allocation method in "Table SSPs".

D: Manually zeroized (key deletion via CN_DESTROY_OBJECT service)

E: Zeroized right after used (memory is wiped with zeros immediately after use)

S: Zeroized on session close (Session close via CN_CLOSE_SESSION, CN_APP_FINALIZE, and CN_CLOSE_PARTITION_SESSIONS services. Please see Table 13 for further details.)

PD: Zeroize all SSPs in the Partition and then delete the User Partition (via CN_DELETE_PARTITION service).

PZ: Zeroize all User SSPs in the Partition (User Partition Regular zeroize via CN_ZEROIZE service)

MZ: Zeroize all Partitions' SSPs, except vendor programmed ones (Master Partition Regular CN_ZEROIZE)

PFZ: Zeroize all SSPs in the Partition (Factory reset via CN_ZEROIZE service with factory-reset as argument with PCO credentials)

MFZ: Zeroize all Partitions' SSPs, including the HSM adapter owner programmed ones (Brings HSM to factory state. Factory reset via CN_ZEROIZE service with factory-reset as argument with MCO credentials.)

VZ: Vendor zeroize (via CN_VENDOR_ZEROIZE service. Zeroizes all SSPs including vendor programmed configuration and CSPs. Makes the module unusable; the module must be sent back to the vendor for re-programming.)

Table 18 – SSPs

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
DRBG Entropy (OCTEON HW RBG)	256-bit	ENT (P) SP 800-90B	ENT (P) SP 800-90B	N/A	SSP generation	In Memory	MFZ, VZ	The entropy input string and seed for the Approved DRBG. Each version of the DRBG has its own DRBG Entropy CSP.
CTR_DRBG Internal State	256-bit	Counter DRBG (SP 800-90Ar1) [#C821]	Derived Entropy from OCTEON HW RBG	N/A	SSP generation	In Memory	PD, PZ, MZ, PFZ, MFZ, VZ	The internal state (V, Key) for the Approved DRBG.
HASH_DRBG Internal State	256-bit	Hash DRBG (SP 800-90Ar1) [#C830]	Derived Entropy from Counter DRBG (SP 800-90Ar1) [#C821])	N/A	SSP generation	In Memory	PD, PZ, MZ, PFZ, MFZ, VZ	The internal state (V, C) for the Approved SHA DRBG.
Manufacturer FIPS Data Encryption Key (MFDEK)	256-bit	AES-CBC (SP 800-38A) [#C819]	CKG SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output (SP 800-90Ar1) [#C821]	No	SSP generation	EEPROM (Plaintext)	VZ	Use: AES 256-bit key used to encrypt manufacturer keys stored in persistent storage of the HSM. Related Keys: FMAK, PAK, MFKBK, OKBK, POKBK

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
HSM Master Partition Master Encryption Key (MMEK)	256-Bit	AES-CBC (SP 800-38A) [#C819]	CKG SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output (SP 800-90AR1) [#C821]	No	SSP generation	EEPROM (Plaintext)	MZ	Use: AES 256-bit key used to encrypt Master Partition CSPs and authentication data stored in persistent storage of the HSM. Related Keys: PMEK
Partition Master Encryption Key (PMEK)	256-bit	AES-CBC (SP 800-38A) [#C819]	CKG SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output) (SP 800-90AR1) [#C821]	No	SSP generation	eMMC flash (Encrypted by MMEK , AES-CBC #C819)	PZ	AES 256-bit key used to encrypt partition CSPs and authentication data stored in persistent storage of the HSM. Related Keys: MMEK
HSM FIPS Master Authentication Key (FMAK)	112-Bit	RSA SigGen (FIPS 186-4) [#C824] RSA SigVer (FIPS 186-4) [#C824], KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output) Counter DRBG (SP 800-90Ar1) [#C821] RSA KeyGen (FIPS 186-4) [#C824]	No	SSP generation	eMMC flash (Encrypted by MMEK, AES-CBC #C819)	VZ	Use: A unique 2048-bit RSA private key. Used to identify the HSM when in the operating in approved mode. Related Keys: PAC, FMAC
Partition Authentication Key (PAK)	112-Bit	RSA SigGen (FIPS 186-4) [#C824] RSA SigVer (FIPS 186-4) [#C824], KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output) Counter DRBG (SP 800-90Ar1) [#C821] RSA KeyGen (FIPS 186-4) [#C824]	No	SSP generation	eMMC flash (Encrypted by PMEK, AES-CBC #C819)	PD	A unique 2048-bit RSA private key used to identify the HSM Partition.

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
SecureAuth Shared Secret (SAZ)	112-Bit	KDF SP 800-108 (KBKDF)[# C839]	KAS (SP 800-56Br2) KAS-IFC OneStep (SP 800-56Br2) KAS-IFC TwoStep (SP 800-56Br2)	No	Key agreement	In Memory (Plaintext)	E	Shared secret Z for SP 800-56Br2 KAS2, using PAK and POAC
PswdEncKey (PEK)	256-Bit	AES CBC [#C819], allowed per IG D.G AES-CBC (SP 800-38A) [#C819]	KAS (SP 800-56Br2) KAS-IFC OneStep (SP 800-56Br2) KAS-IFC TwoStep (SP 800-56Br2)	No	Key agreement	In Memory (Encrypted by PMEK, AES-CBC #C819)	PZ	AES-256 key, for encrypting User passwords during user creation and authentication.
Login Passwords	8 characters	PBKDF (SP 800-132) [#A1196]	N/A	Yes (Import) (Encrypted by PEK, AES-CBC #C819)	Key transport	eMMC flash (Encrypted by PMEK, AES-CBC #C819)	D	String of 8 to 32 alphanumeric characters.
Partition KeyLoading Private Key	256-Bit	KAS-IFC HKDF (SP 800-56Br2) KAS-IFC OneStep (SP 800-56Br2) KAS-IFC TwoStep (SP 800-56Br2) KAS-ECC (KAS) Sp800-56Ar3 [#A1219]	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output ECDSA KeyGen (FIPS 186-4) [#C825] RSA KeyGen (FIPS 186-4) [#C824]	No	SSP generation	In memory (Plaintext)	E	ECC 521-bit or RSA 2048-bit key used in SP 800-56Ar3 C (2,0, ECC DH) or SP 800-56Br2 KAS2 to agree on Z during key loading.
Partition KeyLoading Shared Secret (KLSZ)	256-Bit	KDF SP 800-108 (KBKDF)[#C826]	KAS KDA HKDF (SP 800-56Ar3) KAS-IFC HKDF (SP 800-56Br2) KAS-IFC OneStep (SP 800-56Br2) KAS-IFC TwoStep (SP 800-56Br2)	No	Key agreement	In memory (Plaintext)	E	Shared secret Z for SP 800-56Ar3 C (2,0, ECC DH) or SP 800-56Br2 KAS2.

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
Partition Key Loading Key (KLK)	256-Bit	AES-CBC (SP 800-38A) [#C819], AES-KW (KTS) (SP 800-38F) [#C1263], AES-KWP (KTS) (SP 800-38F) [#C1263], AES-GCM (SP 800-38D) [#A1203], AES-GCM (SP 800-38D) [#C839]	KAS-KDF-HKDF (SP 800-56Ar3) KAS-KDF-OneStep(SP 800-56Ar3) KAS-KDF-TwoStep (SP 800-56Ar3) [KAS-ECC-SSC Cert. #A1220 and KDA Cert. #A1192]	No	Key agreement	eMMC flash (Encrypted by PMEK, AES-CBC #C819)	PZ	256-bit AES key derived from Z, used to decrypt the imported CSPs.
Manufacturer FIPS Key Backup Key (MFKBK)	256-Bit	KDF SP 800-108 (KBKDF) [#C826]	No	Yes (Import) KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	Key transport	eMMC flash (Encrypted by MMEK AES-KW #C1263)	VZ	AES 256-bit key used to derive KBK.
HSM Owner KBK (OKBK)	256-Bit	KDF SP 800-108 (KBKDF) [#C826]	No	Yes (Import) KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	Key transport	eMMC flash (Encrypted by MMEK AES-KW #C1263)	MFZ	AES 256-bit key used to derive KBK.
Partition Owner KBK (POKBK)	256-Bit	KDF SP 800-108 (KBKDF) [#C826]	No	Yes (Import) KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	Key transport	eMMC flash (Encrypted by MMEK AES-KW #C1263)	PFZ	AES 256-bit key used to derive KBK.
HSM Key Backup Key (KBK)	256-Bit	AES-CBC (SP 800-38A) [#C819] AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	KDF SP 800-108 (KBKDF) [#C826]	No	Key derivation/ SSP generation	eMMC flash (Encrypted by MMEK AES-KW #C1263)	MZ	Key used to encrypt/decrypt the Backup Session Key.
Backup Session Key	256-Bit	AES-CBC (SP 800-38A) [#C819] AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	CKG SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output Counter DRBG (SP 800-90Ar1) [#C821]	No	SSP generation	In memory (Plaintext)	E	Key used to backup and restore partition data.

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
Partition Cloning Private Key (PCPK)	256-Bit	KAS-IFC HKDF (SP 800-56Br2) KAS-IFC OneStep (SP 800-56Br2) KAS-IFC TwoStep (SP 800-56Br2) KAS-ECC (KAS) Sp800-56Ar3 [#A1219]	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output ECDSA KeyGen (FIPS 186-4) [#C825] RSA KeyGen (FIPS 186-4) [#C824]	No	SSP generation	In memory (Plaintext)	E	ECC 521-bit or RSA 2048-bit ephemeral Private Key used in SP 800-56Ar3 C (2,0, ECC DH) or SP 800 -56B KAS2 -bilateral -confirmation key agreement to generate shared secret Z. At HSM Partition level, used to establish secure channel for cloning process (to export Partition Masking Key).
Partition Cloning Shared Secret (CSSZ)	256-Bit	KDF SP 800-108 (KBKDF)[#C826]	KAS-ECC (KAS) Sp800-56Ar3 [#A1219]	No	Key agreement	In memory (Plaintext)	E	Shared secret Z for SP 800-56Ar3 C (2,0, ECC DH) or SP 800-56Br2 KAS2 -bilateral -confirmation scheme.
Partition Cloning Session Key (PCSK)	256-Bit	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	KAS-ECC (KAS) Sp800-56Ar3 [#A1219]	No	Key agreement	In Memory (Plaintext)	E	AES 256 key for encryption and decryption of Partition Masking Key.
Partition Cloning Session MAC Key (PCSMK)	256-Bit	HMAC-SHA2-256 (FIPS 198-1) [#C822]	KAS-ECC (KAS) Sp800-56Ar3 [#A1219]	No	Key agreement	In Memory (Plaintext)	E	HMAC SHA2-256 key used for key confirmation during SP 800-56Ar3 key agreement.
Partition Masking Key	256-bit	AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263]	CKG SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output Counter DRBG (SP 800-90Ar1) [#C821]	No	SSP generation	eMMC flash (Encrypted by PMEK, AES-CBC #C819)	PZ	AES-256 key, for key wrapping. Used to import/export CSPs and masked objects.
Asymmetric Private Keys (user keys)	112-256 Bit	ECDSA KeyVer (FIPS 186-4) [#C825] ECDSA SIGGEN (FIPS 186-4) (CVL) [#C825] ECDSA SIGGEN (FIPS 186-4) [#C825] ECDSA SigVer (FIPS 186-4) [#C825] KAS-KDF-HKDF (SP 800-56Ar3) KAS-KDF-OneStep(SP 800-56Ar3) KAS-KDF-TwoStep (SP 800-56Ar3) KAS (SP 800-56Ar3) [#A1220 and C825] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829]	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output DSA KeyGen (FIPS 186-4) [#C823] ECDSA KeyGen (FIPS 186-4) [#C825] RSA KeyGen (FIPS 186-4) [#C824]	Yes (Import/ Export) AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829]	SSP generation/ Key transport	In Memory/ eMMC flash (Plaintext/ Encrypted by PMEK, AES-CBC #C819)	D, S	RSA/DSA/ECDSA/ECDH general purpose keys.

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
		RSA KeyGen (FIPS 186-4) [#C824] RSA SIGGEN (FIPS 186-4) [#A1199] RSA SIGGEN (FIPS 186-4) [#C824] RSA SIGVER (FIPS 186-4) [#A1199] RSA SIGVER (FIPS 186-4) [#A1201] RSA SIGVER (FIPS 186-4) [#C824] DSA KeyGen (FIPS 186-4) [#C823] DSA PQGGen (FIPS 186-4) [#C823] DSA PQGVer (FIPS 186-4) [#C823] DSA SigVer (FIPS 186-4) [#C823] DSA SigVer (FIPS 186-4)) [#C823], RSA Decryption Primitive (CVL) (SP 800-56Br2) [#C839], RSA Signature Primitive (CVL) (FIPS 186-4) [#C839], RSA Decryption Primitive (CVL) (SP 800-56Br2)[#A1200] KAS-ECC-SSC Sp800-56Ar3 [#A1220] KAS KDA HKDF (SP 800-56Ar3) KAS KDA ONESTEP (SP 800-56Ar3) KAS KDA TWOSTEP (SP 800-56Ar3) KAS ANS 9.63 (SP 800-56Ar3) KAS-ECC-SSC Sp800-56Ar3 [#A2161] ECDSA SigGen (FIPS 186-4) (CVL) [#c829] KTS-IFC (KTS) (SP 800-56Br2) [#A1194]		KDA HKDF Sp800-56Cr1 [#A1192] KDA OneStep Sp800-56Cr1 [#A1192] KDA TwoStep Sp800-56Cr1 [#A1192] AES-GCM (KTS) (SP 800-38D) [#A1203] AES-GCM (KTS) (SP 800-38D) [#C839] KTS-IFC (KTS) (SP 800-56Br2) [#A1194]				
Asymmetric Private Session Keys (user keys)	112-256 Bit	ECDSA KeyGen (FIPS 186-4) [#C825] ECDSA KeyVer (FIPS 186-4) [#C825] ECDSA SIGGEN (FIPS 186-4) (CVL) [#C825] ECDSA SIGGEN (FIPS 186-4) [#C825] ECDSA SigVer (FIPS 186-4) [#C825] KAS-KDF-HKDF (SP 800-56Ar3) KAS-KDF-OneStep(SP 800-56Ar3)	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output DSA KeyGen (FIPS 186-4) [#C823]	Yes (Import/Export) AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829]	SSP generation/ Key transport	In Memory Plaintext	D, S	RSA/DSA/ECDSA/ECDH general purpose session keys.

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
		KAS-KDF-TwoStep (SP 800-56Ar3) SP 800 KAS-KDF-ANS9.63 (SP 800-56Ar3) KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] RSA KeyGen (FIPS 186-4) [#A1199] RSA KeyGen (FIPS 186-4) [#C824] RSA SIGGEN (FIPS 186-4) [#A1199] RSA SIGGEN (FIPS 186-4) [#C824] RSA SIGVER (FIPS 186-4) [#A1199] RSA SIGVER (FIPS 186-4) [#A1201] RSA SIGVER (FIPS 186-4) [#C824] DSA KeyGen (FIPS 186-4) [#C823] DSA PQGGen (FIPS 186-4) [#C823] DSA PQGVer (FIPS 186-4) [#C823] DSA SigVer (FIPS 186-4) [#C823] DSA SigVer (FIPS 186-4)) [#C823], RSA Decryption Primitive (CVL) (SP 800-56Br2) [#C839], RSA Signature Primitive (CVL) (FIPS 186-4) [#C839], RSA Decryption Primitive (CVL) (SP 800-56Br2)[#A1200] KAS-ECC-SSC Sp800-56Ar3 [#A1220] KAS KDA HKDF (SP 800-56Ar3) KAS KDA ONESTEP (SP 800-56Ar3) KAS KDA TWOSTEP (SP 800-56Ar3) KAS ANS 9.63 (SP 800-56Ar3) KAS-ECC-SSC Sp800-56Ar3 [#A2161] ECDSA SigGen (FIPS 186-4) (CVL) [#c829] KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	ECDSA KeyGen (FIPS 186-4) [#C825] RSA KeyGen (FIPS 186-4) [#C824]	KDA HKDF Sp800-56Cr1 [#A1192] KDA OneStep Sp800-56Cr1 [#A1192] KDA TwoStep Sp800-56Cr1 [#A1192] AES-GCM (KTS) (SP 800-38D)) [#A1203] AES-GCM (KTS) (SP 800-38D) [#C839] KTS-IFC (KTS) (SP 800-56Br2) [#A1194]				

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establishment	Storage	Zeroization	Use & related keys
Symmetric Keys (user keys)	112-256 Bit	TDES-TKW (#C1263) TDES-ECB (#1311) TDES-CBC (#1311) TDES-CBC (#C1169) TDES-ECB (#C1169) SP 800 KDF SP 800-108 [#C826] KDF SP 800-108 (#C839) KDF SP 800-108 (#A1191) AES [#C819] CBC/ECB [#C819], allowed per IG D.G AES KW (#C1263) AES-KWP (#C1263) AES-CMAC (SP 800-38B)(#C839) AES-CMAC (SP 800-38B)(#A1195) AES-CTR (SP 800-38A) [#C839] AES-GMAC (SP 800-38D) [#C839] AES-CCM (SP 800-38C) [#C839]	CKG SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output Counter DRBG (SP 800-90Ar1) [#C821]	Yes (Import/ Export) AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] KDA HKDF Sp800-56Cr1 [#A1192] KDA OneStep Sp800-56Cr1 [#A1192] KDA TwoStep Sp800-56Cr1 [#A1192] AES-GCM (KTS) (SP 800-38D) [#A1203] AES-GCM (KTS) (SP 800-38D) [#C839] KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	SSP generation/ Key transport	In Memory/ eMMC flash (Plaintext/ Encrypted by PMEK, AES-CBC #C819)	D, S	Triple-DES or AES general purpose keys.
Symmetric Session Keys (user keys)	112-256 Bit	TDES-TKW (#C1263) TDES-ECB (#1311) TDES-CBC (#1311) TDES-CBC (#C1169) TDES-ECB (#C1169) SP 800 KDF SP 800-108 [#C826] KDF SP 800-108 (#C839) KDF SP 800-108 (#A1191) AES[#C819] CBC/ECB [#C819], allowed per IG D.G AES KW (#C1263) AES-KWP (#C1263) AES-CMAC (SP 800-38B)(#C839) AES-CMAC (SP 800-38B)(#A1195) AES-CMAC (SP 800-38B)(#A1190) AES-CTR (SP 800-38A) [#C839] AES-GMAC (SP 800-38D) [#C839] AES-CCM (SP 800-38C) [#C839]	CKG SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output Counter DRBG (SP 800-90Ar1) [#C821]	Yes (Import/ Export) AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] KDA HKDF Sp800-56Cr1 [#A1192] KDA OneStep Sp800-56Cr1 [#A1192] KDA TwoStep Sp800-56Cr1 [#A1192] AES-GCM (KTS) (SP 800-38D) [#A1203] AES-GCM (KTS) (SP 800-38D) [#C839]	SSP generation/ Key transport	In Memory Plaintext	D, S	Triple-DES or AES general purpose session keys.

Key/SSP Name/ Type	**Strength**	**Security Function and Cert. Number**	**Generation**	**Import/ Export**	**Establishment**	**Storage**	**Zeroization**	**Use & related keys**
				KTS-IFC (KTS) (SP 800-56Br2) [#A1194]				
HMAC Keys (user keys)	112-256 Bit	HMAC-SHA2-256 (FIPS 198-1) [#C822], HMAC-SHA-1 (FIPS 198-1) [#C839], HMAC-SHA2-224 (FIPS 198-1) [#C839], HMAC-SHA2-384 (FIPS 198-1) [#C839], HMAC-SHA2-512 (FIPS 198-1) [#C839], HMAC-SHA2-256 (FIPS 198-1) [#C839], HMAC-SHA-1 (FIPS 198-1) [#C822], HMAC-SHA2-224 (FIPS 198-1) [#C822], HMAC-SHA2-384 (FIPS 198-1) [#C822], HMAC-SHA2-512 (FIPS 198-1) [#C822]	CKG SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output Counter DRBG (SP 800-90Ar1) [#C821]	Yes (Import/ Export) AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] KDA HKDF Sp800-56Cr1 [#A1192] KDA OneStep Sp800-56Cr1 [#A1192] KDA TwoStep Sp800-56Cr1 [#A1192] AES-GCM (KTS) (SP 800-38D) [#A1203] AES-GCM (KTS) (SP 800-38D) [#C839] KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	Key transport/ SSP generation	In Memory/eMMC flash (Plaintext/Encrypted by PMEK, AES-CBC #C819)	D, S	HMAC general purpose keys (minimum key size of 160 bits).
HMAC Session Keys (user keys)	112-256 Bit	HMAC-SHA2-256 (FIPS 198-1) [#C822] HMAC-SHA-1 (FIPS 198-1) [#C839], HMAC-SHA2-224 (FIPS 198-1) [#C839], HMAC-SHA2-384 (FIPS 198-1) [#C839], HMAC-SHA2-512 (FIPS 198-1) [#C839], HMAC-SHA2-256 (FIPS 198-1) [#C839], HMAC-SHA-1 (FIPS 198-1) [#C822], HMAC-SHA2-224 (FIPS 198-1) [#C822], HMAC-SHA2-384 (FIPS 198-1) [#C822], HMAC-SHA2-512 (FIPS 198-1) [#C822]	CKG SP 800-133Rev2 Section 6.1 Direct symmetric key generation using unmodified DRBG output Counter DRBG (SP 800-90Ar1) [#C821]	Yes (Import/ Export) AES-KW (KTS) (SP 800-38F) [#C1263] AES-KWP (KTS) (SP 800-38F) [#C1263] KAS-ECC CDH-Component (CVL) (SP 800-56Ar3) [#C829] KDA HKDF Sp800-56Cr1 [#A1192] KDA OneStep Sp800-56Cr1 [#A1192] KDA TwoStep Sp800-56Cr1 [#A1192] AES-GCM (KTS) (SP 800-38D) [#A1203]	Key transport/ SSP generation	In Memory (Plaintext)	D, S	HMAC session general purpose keys (minimum key size of 160 bits).

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
				AES-GCM (KTS) (SP 800-38D) [#C839] KTS-IFC (KTS) (SP 800-56Br2) [#A1194]				
E2E TLS Session ECDH Key	112-256 Bit	KAS TLS (SP 800-56Ar3)	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output ECDSA KeyGen (FIPS 186-4) [#C825]	No	SSP generation	In Memory (Plaintext)	E	Used for key agreement as part of E2E handshake protocol.
TLS Session Symmetric Key Set	112-256 Bit	SP 800 AES-CBC (SP 800-38A) [#C839] AES-GCM (SP 800-38D) [#C839]	KDF TLS (CVL) (SP 800-135r1) [#C840]	No	Key derivation	In Memory (Plaintext)	E	AES 128, 192, 256 or Triple-DES keys used for encrypting TLS sessions.
TLS Session HMAC key	112-256 Bit	HMAC-SHA2-256 (FIPS 198-1) [#C839], HMAC-SHA2-384 (FIPS 198-1) [#C839],	KDF TLS (CVL) (SP 800-135r1) [#C840]	No	Key derivation	In Memory (Plaintext)	E	HMAC key used in SSL session (minimum key size of 160 bits).
E2E TLS Session Symmetric Key Set	112-256 Bit	AES-GCM (SP 800-38D) [#C839]	KDF TLS (CVL) (SP 800-135r1) [#C840]	No	Key derivation	In Memory (Plaintext)	E	AES 128/256-bit Key used for encrypting/decrypting E2E session data.
E2E TLS Session HMAC keys	112-256 Bit	HMAC-SHA2-256 (FIPS 198-1) [#C839], HMAC-SHA2-384 (FIPS 198-1) [#C839],	KDF TLS (CVL) (SP 800-135r1) [#C840]	No	Key derivation	In Memory (Plaintext)	E	HMAC keys used in E2E session.
Manufacturer Firmware Integrity Check Keys	112-Bit	RSA SigVer (FIPS 186-4) [#A1201]	No	No	Pre-loading of a key	eMMC flash (Plaintext)	VZ	RSA 2048-bit public keys used to check the integrity of the SW images booted. The SW image is signed by the manufacturer using a RSA private key. **Note**: This is not an SSP but is included for completeness of the parameters used by the module.

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
Manufacturer Firmware Update Validation Key (MFUVK)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	Pre-loading of a key	eMMC flash (Plaintext)	N/A	RSA 2048-bit public key used to authenticate new SW images uploaded into the module. The SW image is signed by the manufacturer using an RSA private key and the signature is verified before upgrading to the new image using the public key. **Note**: This PSP is considered protected, because it cannot be modified by a user of the module. It is delivered as part of the existing firmware image, which can only be replaced by vendor-signed images.
Manufacturer License Validation Key (MLVK)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	Pre-loading of a key	eMMC flash (Plaintext)	N/A	RSA 2048-bit public key used to authenticate the manufacturer role. **Note**: This PSP is considered protected, because it cannot be modified by a user of the module. It is factory loaded in the module.
Manufacturer Authentication Root Cert. (MARC)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	Pre-loading of a key	eMMC flash (Plaintext)	VZ	RSA 2048-bit public key certificate, used to issue FMAC certificates.
HSM FIPS Master Authentication Certificate (FMAC)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	Pre-loading of a key	eMMC flash (Plaintext)	VZ	RSA 2048-bit public key certificate of FMAK. Used to identify the HSM FIPS operating mode.
SecureBootAuth Public Key	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	Pre-loading of a key	eMMC flash (Plaintext)	MFZ	RSA 2048-bit public key used to verify authenticity of the host system.
HSM/Adapter Owner Trust Anchor Certificate (AOTAC)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	SSP Entry	eMMC flash (Plaintext)	MFZ	RSA 2048-bit public key certificate used as trust anchor of MCO.
HSM/Adapter Owner Authentication Certificate (AOAC)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	SSP Entry	eMMC flash Plaintext)	MFZ	RSA 2048-bit public key certificate of FMAK. Used to identify the HSM owner.

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
Partition Authentication Certificate (PAC)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output RSA KeyGen (FIPS 186-4) [#C824]	No	SSP Entry/ SSP generation	eMMC flash (Plaintext)	PD	RSA 2048-bit public key certificate of PAK. Used to identify the Partition.
Partition Owner Trust Anchor Certificate (POTAC)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	SSP Entry	eMMC flash (Plaintext)	PFZ	RSA 2048-bit public key certificate used as trust anchor of PCO.
Partition Owner Authentication Certificate (POAC)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	SSP Entry	eMMC flash (Plaintext)	PFZ	RSA 2048-bit public key certificate of PAK. Used to identify the Partition owner.
Partition Cloning Initiator Public Key	256-Bit	ECDSA SigVer (FIPS 186-4) [#C825] KAS-ECC (KAS) **Sp800-56Ar3** [#A1219]	No	No	SSP Entry	In memory (Plaintext)	E	ECC 521-bit ephemeral public key used in SP 800-56Ar3 C (2,0, ECC DH) key agreement or RSA 2048-bit ephemeral public key used in SP 800-56Br2 KAS2 -bilateral -confirmation key agreement to generate shared secret Z.
Partition Cloning Responder Public Key	256-Bit	ECDSA SigVer (FIPS 186-4) [#C825] KAS-ECC (KAS) Sp800-56Ar3 [#A1219]	No	No	SSP Entry	In memory (Plaintext)	E	ECC 521-bit ephemeral public key used in SP 800-56Ar3 C (2, 0, ECC DH) key agreement or RSA 2048-bit ephemeral public key used in SP 800-56Br2 KAS2 -bilateral -confirmation key agreement to generate shared secret Z.
Host PswdEncKeyPublicKey	112-Bit	RSA SigVer (FIPS 186-4) [#C824] KAS-IFC HKDF (SP 800-56Br2) KAS-IFC OneStep (SP 800-56Br2) KAS-IFC TwoStep (SP 800-56Br2)	No	No	SSP entry	In Memory (Plaintext)	E	RSA 2048-bit public key loaded by the host to be used SP 800-56Br2 key agreement to generate PswdEncKey.
Two-Factor Authentication Public Key or MofN Authentication Key (2FAMofNPubK)	112-Bit	RSA SigVer (FIPS 186-4) [#C824]	No	No	SSP Entry	eMMC flash (Encrypted by PMEK, AES-CBC #C819)	PZ	RSA 2048-bit public key used to verify signature on encrypted passwords during user creation and login and/or to verify signatures on MofN authentication tokens.
E2E Client Authentication Public key (CAPubK)	112-256 Bit	RSA SigVer (FIPS 186-4) [#C824] ECDSA SigVer (FIPS 186-4) [#C825]	No	No	SSP Entry	eMMC flash (Plaintext)	PZ	RSA or EC public key of approved modulus or curveId to allow E2E/TLS client authentication in E2E/TLS handshake

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
Adapter Owner Attestation Public key (AOAPubK)	112-256 Bit	RSA SigVer (FIPS 186-4) [#C824] ECDSA SigVer (FIPS 186-4) [#C825]	No	No	SSP Entry	eMMC flash (Plaintext)	PZ	RSA or EC public key of approved modulus or curveId to allow HSM/Adapter Owner to authenticated with signature verification with this public key and perform FW image update
User Public Keys (user keys)	112-256 Bit	ECDSA SigVer (FIPS 186-4) [#C825] RSA SigVer (FIPS 186-4) [#C824] KTS-IFC (KTS) (SP 800-56Br2) [#A1194] DSA KeyGen (FIPS 186-4) [#C823] DSA PQGGen (FIPS 186-4) [#C823] DSA PQGVer (FIPS 186-4) [#C823] DSA SigVer (FIPS 186-4) [#C823] DSA SigVer (FIPS 186-4) [#C823] KAS-ECC-SSC Sp800-56Ar3 [#A1220] KAS KDA HKDF (SP 800-56Ar3) KAS KDA ONESTEP (SP 800-56Ar3) KAS KDA TWOSTEP (SP 800-56Ar3) KAS ANS 9.63 (SP 800-56Ar3) KAS-ECC-SSC Sp800-56Ar3 [#A2161] ECDSA SigVer (FIPS 186-4) [#C829]	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output DSA KeyGen (FIPS 186-4) [#C823] ECDSA KeyGen (FIPS 186-4) [#C825] RSA KeyGen (FIPS 186-4) [#C824]	Yes (Import Plaintext)	SSP entry/ SSP generation	eMMC flash (Encrypted by PMEK, AES-CBC #C819)	D	RSA/DSA/ECDSA/ECDH public keys.

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establish-ment	Storage	Zeroization	Use & related keys
User Public Session Keys (user keys)	112-256 Bit	ECDSA SigVer (FIPS 186-4) [#C825] RSA SigVer (FIPS 186-4) [#C824] DSA KeyGen (FIPS 186-4) [#C823] DSA PQGGen (FIPS 186-4) [#C823] DSA PQGVer (FIPS 186-4) [#C823] DSA SigVer (FIPS 186-4) [#C823] DSA SigVer (FIPS 186-4) [#C823] KAS-ECC-SSC Sp800-56Ar3 [#A1220] KAS KDA HKDF (SP 800-56Ar3) KAS KDA ONESTEP (SP 800-56Ar3) KAS KDA TWOSTEP (SP 800-56Ar3) KAS ANS 9.63 (SP 800-56Ar3) KAS-ECC-SSC Sp800-56Ar3 [#A2161] ECDSA SigVer (FIPS 186-4) [#C829] KTS-IFC (KTS) (SP 800-56Br2) [#A1194]	CKG SP 800-133Rev2 Section 5.2 Asymmetric key establishment, key generation using unmodified DRBG output DSA KeyGen (FIPS 186-4) [#C823] ECDSA KeyGen (FIPS 186-4) [#C825] RSA KeyGen (FIPS 186-4) [#C824]	No	SSP entry/ SSP generation	In Memory (Plaintext)	D, S	RSA/DSA/ECDSA/ECDH public session keys.

Table 19 – Non-Deterministic Random Number Generation Specification

Entropy sources	Minimum number of bits of entropy	Details
OCTEON HW RBG	Gathers 256-bit security strength of entropy	The OCTEON II HW unit only attributes entropy to random bits generated from the 8-free running oscillators, from a total of 128-free running oscillators. The generated random bits are run through software/firmware health tests (APT and RCT).

9.2 *Definition of Session Keys*

The cryptographic module supports the generation/import/export of user keys that are bound to a session and are termed as session keys. The following points apply to session keys:

- Session keys are stored in RAM and are lost across reboots.
- Session key access is restricted to an application in which it is created. PCU can share the session keys with other users, so that other sessions can use it.
- Every session in an application will have access to the keys created by every other session in the same application.
- When a session is closed, the session keys created by that session get zeroized (Session Keys memory being overwritten with zeros before release). If the key is shared, then it will be deleted only after closing all the sessions sharing this key

10 Self-Tests

This section documents the security rules enforced by the cryptographic module to implement the self-test requirements of this FIPS 140-3 Level-3 module.

The module always executes the self-tests without operator intervention regardless of approved or non-approved mode or any other configuration.

Failure of any of these tests causes the module to go into an error state and all future commands received are rejected by the module.

The module needs to be reset to recover from the situation. Data output except for status log messages is inhibited during self-tests, zeroization, and error states. Status information does not contain CSPs or sensitive data.

The conditional cryptographic algorithm self-tests (CAST) run periodically. The periodicity is configurable by the MCO and by default runs for every 8 hours. These self-tests can be triggered by the operator via MCO role. Please refer to Section 11.3 for guidance on how to initiate these tests. The execution of CASTs causes a momentary (less than a second) service interruption.

The operator is capable of commanding the module to perform the pre-operational and conditional self-tests by cycling power or resetting the module.

The voltage and temperature monitoring is performed continuously by the module every 30 seconds.

The cryptographic module performs the following pre-operational and conditional self-tests:

- Pre-Operational Self-Tests (7.10.2):
  - Pre-operational software/firmware integrity test (7.10.2.2):
    - CRC-32 Integrity tests
    - Firmware Integrity Tests (#A1201, RSA 2048-bit SHA2-256 signature verification)
  - Pre-operational bypass test (7.10.2.3):
    - None
  - Pre-operational Critical Functions Tests (7.10.2.4): The module runs the following Critical Functions Tests which are required to ensure the correct functioning of the device.
    - Power On Memory Test
    - EEPROM Test
    - NOR Flash Test

- − Nitrox Chips Tests
- − Temperature monitor test
- − Voltage monitor test

- Conditional Self-Tests (7.10.3):
  - Conditional cryptographic algorithm self-test (7.10.3.2):

    a. NITROX Library
    - FIPS 186-4 ECDSA SigGen KAT (#C829, P256 using SHA-1, SHA2-256, SHA2-384, SHA2-512, SHS#1780)
    - FIPS 186-4 ECDSA SigVer KAT (#C829, P256 using SHA-1, SHA2-256, SHA2-384, SHA2-512, SHS#1780)
    - FIPS 186-4 RSA Signature Primitive KAT (#C839, 2048bit)
    - SP 800-108 KDF CMAC in Counter mode KAT (#C839, AES 128bit Key)
    - SP 800-108 KDF HMAC in Counter mode KAT (#C839, HMAC-SHA2-256)
    - SP 800-135r1 KDF TLS KAT (#C840, TLS 1.0/1.1)
    - SP 800-135r1 KDF TLS KAT (#C840, TLS 1.2 HMAC-SHA2-256)
    - SP 800-38A AES-CBC Decrypt KAT (#C839, 128bit Key)
    - SP 800-38A AES-CBC Encrypt KAT (#C839, 128bit Key)
    - SP 800-38A Triple-DES-CBC Decrypt KAT (#1311, Triple DES 192bit Key)
    - SP 800-38A Triple-DES-CBC Encrypt KAT (#1311, Triple DES 192bit Key)*
    - SP 800-38C AES-CCM Decrypt KAT (#C839, 128bit Key)
    - SP 800-38C AES-CCM Encrypt KAT (#C839, 128bit Key)
    - SP 800-38D AES-GCM Decrypt KAT (#C839, #A1203, 128bit Key)
    - SP 800-38D AES-GCM Encrypt KAT (#C839, #A1203, 128bit Key)
    - SP 800-56Ar3 KAS-ECC CDH-Component KAT (#C829, P256 and P384)
    - SP 800-56Br2 KTS-IFC OAEP Decrypt KAT (#A1194, 2048-bit, 3072-bit and 4096-bit)
    - SP 800-56Br2 KTS-IFC OAEP Encrypt KAT (#A1194, 2048-bit, 3072-bit and 4096-bit)
    - SP 800-56Br2 RSA Decryption Primitive KAT (#C839, 2048bit)
    - SP 800-56Br2 RSA Encryption Primitive KAT (2048bit)
    - SP 800-90Ar1 Hash DRBG (instantiate/generate/reseed) KAT (#C830, SHA2-512)

    b. OpenSSL Library
    - FIPS 186-4 DSA SigGen KAT (#C823, 2048bit, SHA2-256)
    - FIPS 186-4 DSA SigVer KAT (#C823, 2048bit, SHA2-256)
    - FIPS 186-4 ECDSA PKV KAT (#C825, P256)
    - FIPS 186-4 ECDSA SigGen KAT (#C825, P256 with SHA2-256, SHA2-384, SHA2-512 #C820)
    - FIPS 186-4 ECDSA SigVer KAT (#C825, P256 with SHA-1, SHA2-256, SHA2-384, SHA2-512 #C820)
    - FIPS 186-4 RSA SigGen KAT (#C824, #A1199, 2048bit)
    - FIPS 186-4 RSA SigVer KAT (#C824, #A1199, 2048bit)
    - FIPS 202 SHA3-512 KAT (#A1197)
    - SP 800-108 KDF HMAC KAT (#C826 and #C822 with HMAC-SHA2-256)
    - SP 800-132 PBKDF KAT (#A1196, SHA-1, SHA2-256, SHA2-512)
    - SP 800-135r1 KDF ANS 9.63 KAT (#C825, SHA2-224)
    - SP 800-38A AES-CBC Decrypt KAT (#C819, 128bit Key)
    - SP 800-38A AES-CBC Encrypt KAT (#C819, 128bit Key)
    - SP 800-38B AES CMAC KAT (#A1195, 128bit Key)

- SP 800-38B Triple-DES CMAC KAT (#A1198, 192bit Key)
- SP 800-56Br2 RSA Decryption Primitive KAT (#A1200, 2048bit)
- SP 800-56Br2 RSA Encryption Primitive KAT (2048bit)
- SP 800-90Ar1 Counter DRBG (instantiate/generate/reseed) KAT (#C821, AES-256bit Key)

c. Others
- FIPS 186-4 RSA SigVer KAT (#A1201, RSA 2048-bit SHA2-256 signature verification)
- SP 800-108 KDF CMAC (counter) KAT (#A1191, AES-128bit Key)
- SP 800-108 KDF CMAC (counter) KAT (#A1191, Triple-DES 192bit Key)*
- SP 800-38B AES-CMAC (hybrid) KAT (#A1190, 128bit Key)
- SP 800-38F AES-KW Key Unwrap KAT (#C1263, LiquidSecurity Keywrap, 256-bit Key)
- SP 800-38F AES-KW Key Wrap KAT (#C1263, LiquidSecurity Keywrap, 256-bit Key)
- SP 800-38F Triple-DES-KW Key Unwrap KAT (#C1263, LiquidSecurity Keywrap, Triple-DES CBC #C1169, 192bit Key)
- SP 800-38F Triple-DES-KW Key Wrap KAT (#C1263, LiquidSecurity Keywrap, Triple-DES CBC #C1169, 192bit Key)*
- SP 800-38G AES-FF1 encrypt KAT (#A1189, 128bit Key)*
- SP 800-56Ar3 KAS-ECC KAT (#A1219, P521 and HMAC-SHA2-512)
- SP 800-56Ar3 KAS-ECC-SSC KAT (#A1220, P521)
- SP 800-56Ar3 required assurances**
- SP 800-56Br2 KAS-IFC-SSC KAT (#A1193, 3072bit Key – CRT, SHA2-384)
- SP 800-56Cr1 KAS-KDF One-Step KAT (#A1192, SHA2-224 and HMAC-SHA2-256)
- SP 800-56Cr1 KAS-KDF Two-Step KAT (#A1192, SHA2-224 and HMAC-SHA2-256)
- SP 800-90Ar1 Counter DRBG (instantiate/generate/reseed) health tests (#C821 and #C830)
- SP 800-90B Health Tests (RCT/APT)
- SP 800-90B fault detection startup health tests

*Algorithm only used for self-test.
**All SP 800-56Ar3 implementations.

Note: "CN_INVOKE_FIPS" will execute all the listed CASTs.

- Conditional pair-wise consistency test (7.10.3.3):
  - ECDSA Pairwise Consistency Test (#C825) at the time of key generation and import
  - RSA Pairwise Consistency Test (#C824, #A1199) at the time of key generation and import
  - DSA Pairwise Consistency Test (#C823) at the time of key generation and import
- Conditional software/firmware load test (7.10.3.4):
  - Firmware load test (RSA Signature Verification) (#C824, 2048bit Key, SHA2-256)
- Conditional manual entry test (7.10.3.5):
  - None
- Conditional bypass test (7.10.3.6):
  - None
- Conditional critical functions test (7.10.3.7):
  - Temperature monitoring test.
  - Voltage monitoring test.
  - PKCS Sign and verify (#C824) Mod exp CRT with private key and verify with public key

10.1 LED Error Pattern for Self-Test Failure

On successful completion of the self tests, the D10 Green LED remains in the “ON” state. Blinking indicates failures on the HSM. If the LED remains in the permanent glow, the card’s state is fine. All blinks are 200ms ON and 200ms OFF. Blink delay time gap is 1000ms.

These tests map to the tests listed in section 10 Self-Tests.

Table 20 – LED Flash Pattern for Errors

FIPS Test	LED Pattern					
	LED No.	Color	Red	Green	Blue	Blinks
N3 AES-CBC Encrypt/Decrypt	D12	Red	Y	N	N	2
N3 AES-GCM Encrypt	D12	Red	Y	N	N	2
N3 AES-GCM Decrypt	D12	Red	Y	N	N	2
N3 AES-CCM Encrypt	D12	Red	Y	N	N	2
N3 AES-CCM Decrypt	D12	Red	Y	N	N	2
N3 AES-CMAC KDF	D12	Red	Y	N	N	3
N3 HMAC KDF	D12	Red	Y	N	N	3
N3 TLS KDF	D12	Red	Y	N	N	3
N3 Triple-DES-CBC Encrypt/Decrypt	D12	Red	Y	N	N	4
N3 RSASP1	D12	Red	Y	N	N	5
N3 RSA Enc and Dec	D12	Red	Y	N	N	5
RSA OAEP KTS (FW + NITROX)	D12	Red	Y	N	N	5
AES CMAC (FW + NITROX)	D12	Red	Y	N	N	6
N3 HMAC	D12	Red	Y	N	N	7
N3 ECC CDH	D12	Green	N	Y	N	2
N3 ECDSA Sig Verify	D12	Red	N	N	Y	5
N3 DRBG SHA	D12	Green	N	Y	N	3
N3 SHA1	D12	Green	N	Y	N	4
N3 SHA-256 and SHA-512	D12	Green	N	Y	N	4
OpenSSL AESCBC Encrypt/Decrypt	D12	Blue	N	N	Y	2
OpenSSL DSA Sign/Verify	D12	Green	N	N	Y	3
OpenSSL DRBG CTR	D12	Blue	N	N	Y	7
OpenSSL ECDSA PKV	D12	Blue	N	N	Y	4
OpenSSL ECDSA Sign/Verify KAT	D12	Blue	N	N	Y	4
OpenSSL RSA Sign/Verify KAT	D12	Blue	N	N	Y	5
OpenSSL RSA Encrypt/Decrypt	D12	Blue	N	N	Y	5
OpenSSL HMAC KDF	D12	Blue	N	N	Y	6

FIPS Test	LED Pattern					
	LED No.	Color	Red	Green	Blue	Blinks
OpenSSL X963 KDF	D12	Blue	N	N	Y	6
OpenSSL CMAC KDF	D12	Blue	N	N	Y	6
AES KeyWrap	D12	Blue	N	N	Y	2
AES KeyUnwrap	D12	Blue	N	N	Y	2
Triple-DES KeyWrap	D12	Blue	N	N	Y	8
SP 800-56Ar3 KAS	D12	Blue	N	N	Y	2
OpenSSL SHA-1	D12	Green	N	Y	N	5
OpenSSL SHA-256 and SHA-512	D12	Green	N	Y	N	5
PBKDF (SP 800-132) [#A1196]	D12	Green	N	Y	N	6
AES FF1	D12	Green	N	Y	N	7
OpenSSL HMAC	D12	Green	N	Y	N	8
SP 800-56Cr2 KDFs (OneStep and Two-step)	D12	Blue	N	N	Y	6
SP 800-56Ar3 ECDH + SSC	D12	Blue	N	N	Y	9
SP 800-56Ar3 SSC	D12	Blue	N	N	Y	9
OpenSSL Triple-DES CMAC	D12	Blue	N	N	Y	8
ECDSA pair wise consistency test	D12	Blue	N	N	Y	4
RSA pair wise consistency test	D12	Blue	N	N	Y	5
DSA pair wise consistency test	D12	Green	N	Y	N	1
Firmware Power-on Tests						
NITROX device file creation	D14	Red	Y	N	N	1
NITROX driver load fails	D14	Red	Y	N	N	2
NITROX micro code load fails	D14	Red	Y	N	N	3
NITROX pot test failures	D14	Red	Y	N	N	4
Database creation fails	D14	Red	Y	N	N	5
Mgmt daemon has not started successfully	D14	Red	Y	N	N	6
HW RBG for firmware	D12	Blue	N	N	Y	3
Other Firmware States						
HSM Boot stage 1	D10	Red	Y	N	N	No blink
FW integrity Failure state	D10/D12/D14	Red	R	N	N	30 sec on and reboot
HSM Boot stage 2	D10	Red	Y	N	N	Blink (definite)

FIPS Test	LED Pattern					
	LED No.	Color	Red	Green	Blue	Blinks
HSM Boot stage 3(SE-APP initialized Linux handshake not done)	D10	Violet	Y	N	Y	No blink
HSM Linux handshake done, host driver handshake not done	D10	Violet	Y	N	Y	Infinite
HSM PF driver handshake complete	D10	Green	N	Y	N	No blink
HSM admin driver handshake done	D10	Blue	N	N	Y	No blink

11 Life-Cycle Assurance

11.1 Secure Installation, Initialization, Startup, and Operation of the Module

Before installing the HSM, the customer verifies the following:

- The ESD bag in which the HSM is placed in the shipping container is sealed and has not been tampered with.
- The part number and other information included in the label on the HSM matches the label on the shipping bag.
- There is no evidence of physical tampering on the HSM itself.

After this is verified, the customer can physically insert the HSM into the PCIe on the host server.

The host must meet the following minimum requirements:

- x86
- Low-profile PCIe Gen 4x8
- SR-IOV support enabled

After the HSM is physically installed, the LiquidSecurity driver and utilities that communicate with the HSM are installed on the host using the Linux make utility. The user must be logged in as root to perform the installation.

The HSM owner then completes the following steps to claim ownership of the HSM and sets the “fips_state” flag:

1. Loads the driver (command: `insmod <driver.ko>`).
2. Invokes Cfm2Master Utility and logs in as default crypto officer (CO) and initializes the HSM.

   For example:

   ```
   Command: initHSM -p <CO password> -sO <CO user name> -fips_state 2
   ```

   As part of initializing the HSM:

   - The Master Crypto Officer is created with username/password (see Table 11, Table 12, and Table 10 for a description of this role, authentication requirements, and service access).
   - The "fips_state”flag is set on the HSM (non-Approved[“0”], Approved with single- or dual-factor authentication[“2”], or Approved with dual-factor authentication[“3”] required).

After the HSM is initialized, the restrictions detailed in Table 10 are enforced and the default user no longer has access to the HSM.

As a final step, the HSM owner ensures that only they can be authenticated for backup/restore and cloning operations on the HSM by loading the adapter owner certificates (AOTAC and AOAC) on the HSM and generating the HSM owner fixed backup key (OKBK). These steps are taken by the MCO using Cfm2MasterUtil with below given commands.

Command syntax for AOTAC, AOAC, and OKBK storage, there by HSM owner claiming the HSM ownership:

```
Command: storeCert -s <cert-type> -f <Adapter Owner Trust Anchor Cert>.crt
Command: storeCert -s <cert-type> -f <Adapter Owner Auth Cert>.crt
Command: storeMCOFixedKey -f <path>/<OKBK-file> -k <path>/HSMOwner.key
```

11.2 Maintenance Requirements

N/A

11.3 Administrator and Non-Administrator Guidance

The specific tasks that can be performed by users on the HSM are strictly limited by their user role (see Table 10 for details).

A user of the module can query the module's device information, operational parameters, operating mode by invoking the command getHSMInfo from the Cfm2MasterUtil. "FIPS state" member of the output will indicate the module to be in one of the following modes:

- Approved mode with 1FA ("2")
- Approved mode with 2FA ("3")
- Non-Approved mode ("0")
- Zeroized ("-1")

Example command syntax:

```
    Command: getHSMInfo
```

When the module is in "zeroized" state, default user credentials can be used to authenticate, where username is "liquidsecurity" and password is "password".

The instances of zeroization mentioned below are executed through the utilities.

There are different types of zeroization which can be executed through utilities:

- zeroizeHSM --> example command: `Cfm2MasterUtil singlecmd zeroizeHSM`.

  This zeroizeHSM will delete all the Partitions, which in turn will zeroize all partition CSPs, including the temporary SSPs.
  - This maps to CN_ZEROIZE service.
  - zeroizeHSM along with option "-factory_reset" can be used by operator to zeroize all non-vendor specific CSPs. This maps to CN_ZEROIZE service with –factory_reset option.
- Vendor zeroizeHSM can be used to zeroize all SSPs in the module.

  To perform a vendor zeroize, the operator must log in as Crypto Officer. Without the credentials of the crypto officer, the vendor zeroize can't be executed.
  - This maps to CN_VENDOR_ZEROIZE service.
  - example command: `Cfm2MasterUtil singlecmd zeroizeHSM -vendor`.

Notes:

- Temporary SSPs (e.g., session keys and Integrity test values) are forcefully memset to 0 during session close, application close, partition deletion and zeroization of the partition or the HSM.

- Reboot is a power cycle operation which will lead to the zeroization of temporary SSPs like session Keys.
- Zeroization of a partition or HSM can take a few minutes to complete . The zeroization request execution is completed only after zeroization of required CSPs is completed. User is notified about delay with a notification log as depicted below. The Operator must remain in control of the module while the zeroization process is executing.

  For example (the below ouptut is executed through Marvell provided driver utilities):

  ```
  Cfm2MasterUtil singlecmd zeroizeHSM
           Version info, Driver Version: 2.09.07.00, SDK API Version: 2.09.07.00
           Cfm2AppInitWithExtNonce () returned app id : 00de8000
           Cfm2OpenSession2() returned 0x00 : HSM Return: SUCCESS
   Command: zeroizeHSM
               Successful zeroization of HSM will reboot the HSM and
               Host-HSM handshake will be re-done.
               Please wait, this may take few minutes.
          Cfm2ZeroizeHSM returned: 0x00 : HSM Return: SUCCESS
          Current FIPS mode is: ffffffff
  ```

  Note: Unsigned `ffffffff` indicates zeroized, which is –1.
- DRBG context: On zeroizeHSM command DRBG-related contexts are reset. This means that the internal state of the context with respect to DRBG becomes unusable.
- In case the power is lost unexpectedly and if the TDES keys are permanent keys, then the module can continue to use the TDES keys for encryption after the module restart as well, until the encryption limit is reached.

On a routine basis, the MCO can verify that the HSM is correctly operating in approved mode by providing MCO credentials and invoking the command “fipsTest” from the Cfm2MasterUtil utility. The fipsTest utility invokes CN_INVOKE_FIPS service to perform the CAST.

Example command syntax:

```
Command: fipsTest
```

The following compliance-specific conditions are applicable to the HSM module:

- There are no restrictions on which keys or CSPs are zeroized by the vendor zeroization service (CN_VENDOR_ZEROIZE).
- The module does not support a maintenance interface or role.
- The module does not support bypass capabilities.
- The module does not support manual key entry.
- The module does not enter or output plaintext CSPs.
- The module has no CSP feedback to operators. The module does not output intermediate key values part of any operation.
- The cryptographic module clears previous authentications on power cycle. The module does not let access SSPs between approved/non-approved and requires zeroization of the HSM/partition.
- When the module has not been placed in a valid role, the operator shall not have access to any cryptographic services.

11.4 User Guidance

AES GCM IV Restoration upon Power Cycle of Module:

In case the module's power is lost and then restored, the module will establish new sessions which will generate new IVs. For an E2E (TLS) session, this will result in fresh handshake and so new AES-GCM Keys and IVs will be established for the new session.

12 Mitigation of Other Attacks

No mitigation of other attacks is implemented by the module.

13 References

1. NIST Key Wrap Specification SP 800-38F, December 2012.
2. NIST Special Publication 800-38D November 2007.
3. NIST Special Publication 800-56A rev3, April 2018.
4. NIST Special Publication 800-56B rev2, March 2019.
5. NIST Special Publication 800-56C rev2, August 2020.
6. NIST Special Publication 800-57 Part-1 rev5, May 2020.
7. FIPS PUB 186-4, Digital Signature Standard (DSS), July 2013.
8. FIPS PUB 140-3, FIPS Publication 140-3 Security Requirements for Cryptographic Modules.
9. NIST Special Publication 800-90A rev1, June 2015
10. NIST Special Publication 800-90B, January 2018
11. Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program
12. NIST Special Publication 800-131Ar2, March 2019.
13. NIST Special Publication 800-133 rev2, June 2020
14. NIST Special Publication 800-108, October 2009
15. NIST Special Publication 800-135 Revision 1, December 2011
16. CNL35xx-NFBE-Driver-SDK_UserGuide-2.08-01-Rev3
17. NIST Special Publication 800-52 rev2, August 2019.

14 Definitions and Acronyms

MCO – Master Crypto Officer

PCO – Partition Crypto Officer

PCU – Partition Crypto User

HSM – Hardware Security Module

KBK – Key Backup Key

KLK – Key Loading Key

KAT – Known Answer Test

KAS – Key Agreement Scheme

SR-IOV – Single Root I/O Virtualization

2FA – 2 Factor Authentication

CAST – Cryptographic Algorithm Self-Tests