

Fortanix, Inc.

Fortanix DSM Appliance

FIPS 140-3 Non-Proprietary Security Policy



Fortanix®

Table of Contents

1 General	6
1.1 Overview	6
1.2 Security Levels	6
2 Cryptographic Module Specification	6
2.1 Description	6
2.2 Tested and Vendor Affirmed Module Version and Identification	11
2.3 Excluded Components	12
2.4 Modes of Operation	13
2.5 Algorithms	13
2.6 Security Function Implementations	17
2.7 Algorithm Specific Information	23
2.8 RBG and Entropy	24
2.9 Key Generation	24
2.10 Key Establishment	25
2.11 Industry Protocols	25
3 Cryptographic Module Interfaces	25
3.1 Ports and Interfaces	25
3.2 Trusted Channel Specification	26
4 Roles, Services, and Authentication	27
4.1 Authentication Methods	27
4.2 Roles	32
4.3 Approved Services	33
4.4 Non-Approved Services	76
4.5 External Software/Firmware Loaded	76
5 Software/Firmware Security	76
5.1 Integrity Techniques	76
5.2 Initiate on Demand	77
6 Operational Environment	77
6.1 Operational Environment Type and Requirements	77
7 Physical Security	77
7.1 Mechanisms and Actions Required	77
7.5 EFP/EFT Information	80
7.6 Hardness Testing Temperature Ranges	80
7.7 Additional Information	81
8 Non-Invasive Security	81

9 Sensitive Security Parameters Management.....	81
9.1 Storage Areas	81
9.2 SSP Input-Output Methods.....	81
9.3 SSP Zeroization Methods	82
9.4 SSPs	83
9.5 Transitions.....	99
10 Self-Tests.....	99
10.1 Pre-Operational Self-Tests	99
10.2 Conditional Self-Tests.....	100
10.3 Periodic Self-Test Information.....	106
10.4 Error States	108
10.5 Operator Initiation of Self-Tests	109
10.6 Additional Information	109
11 Life-Cycle Assurance	109
11.1 Installation, Initialization, and Startup Procedures.....	109
11.2 Administrator Guidance	110
11.3 Non-Administrator Guidance.....	110
11.4 Design and Rules	110
11.6 End of Life	111
12 Mitigation of Other Attacks	111

List of Tables

Table 1: Document History	5
Table 2: Security Levels	6
Table 3: Tested Module Identification – Hardware	12
Table 4: Modes List and Description	13
Table 5: Approved Algorithms	17
Table 6: Vendor-Affirmed Algorithms	17
Table 7: Security Function Implementations	23
Table 8: Entropy Certificates	24
Table 9: Entropy Sources	24
Table 10: Ports and Interfaces	26
Table 11: Authentication Methods	31
Table 12: Roles	33
Table 13: Approved Services	75
Table 14: Mechanisms and Actions Required	78
Table 15: EFP/EFT Information	80
Table 16: Hardness Testing Temperatures	80
Table 17: Storage Areas	81
Table 18: SSP Input-Output Methods	82
Table 19: SSP Zeroization Methods	83
Table 20: SSP Table 1	91
Table 21: SSP Table 2	99
Table 22: Pre-Operational Self-Tests	100
Table 23: Conditional Self-Tests	106
Table 24: Pre-Operational Periodic Information	106
Table 25: Conditional Periodic Information	108
Table 26: Error States	109

List of Figures

Figure 1: Series 2 Front View (FX2200-II-TN-F)	7
Figure 2: Series 3 Front View (FX3400-F)	7
Figure 3: Series 3 Front View (FX3500-F)	7
Figure 4: Series 2 Copper Rear View (FX2200-II-TN-F)	8
Figure 5: Series 2 Fiber Rear View (FX2200-II-SXN-F)	8
Figure 6: Series 3 Fiber Rear View (FX3400-F)	8
Figure 7: Series 3 Copper Rear View (FX3400-T)	9
Figure 8: Series 3 Fiber Rear View (FX3500-F)	9
Figure 9: Series 3 Copper Rear View (FX3500-T)	9
Figure 10: Hardware Block Diagram	10
Figure 11: Firmware Block Diagram	11
Figure 12: Tamper Evident Label Positions on FX2200-II-TN-F, FX2200-II-SXN-F, FX3400-T, FX3400-F, FX3500-T and FX3500-F	78
Figure 13: Tamper Evident Label Positions on FX3400-T, FX3400-F, FX3500-T and FX3500-F	79
Figure 14: Tamper Evident Label	79
Figure 15: Tamper Evident Label Closeup	80

Author	Title	Date	Version	Description
Richard Souleles	Security Certification Engineer	3/31/2026	1.0	Initial public release

Table 1: Document History

1 General

1.1 Overview

This document is a FIPS 140-3 non-proprietary Security Policy for the Fortanix DSM appliance. The Fortanix DSM appliance is the building block for running Fortanix Data Security Manager™ (DSM), a unified HSM and Key Management solution. With DSM, you can securely generate, store, and use cryptographic keys and certificates, as well as secrets, such as passwords, API keys, tokens, or any blob of data. DSM ensures that you remain in complete control over your keys and secrets. Your business-critical applications and containers can integrate with DSM using legacy cryptographic interfaces or using its native RESTful interface. DSM provides control of and visibility into your key management operations using a centralized web-based UI with enterprise level access controls and comprehensive auditing. DSM is built to scale horizontally and geographically as your demand for managing your keys and secrets increases, while providing automated load-balancing and high availability.

This document can be freely distributed in its entirety without modification

1.2 Security Levels

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
	Overall Level	3

Table 2: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

See section 1.1 Overview for a description of the purpose and use of the module.

Module Type: Hardware

Module Embodiment: Multi-Chip Standalone

Module Characteristics:

Cryptographic Boundary:

The tested cryptographic boundary of the module is the enclosure that contains components of the module. The strong enclosure of the cryptographic module is opaque within the visible spectrum. The module uses tamper evident labels to provide evidence of tampering. The module contains tamper response and zeroization circuitry.



Figure 1: Series 2 Front View (FX2200-II-TN-F)



Figure 2: Series 3 Front View (FX3400-F)



Figure 3: Series 3 Front View (FX3500-F)



Figure 4: Series 2 Copper Rear View (FX2200-II-TN-F)



Figure 5: Series 2 Fiber Rear View (FX2200-II-SXN-F)



Figure 6: Series 3 Fiber Rear View (FX3400-F)



Figure 7: Series 3 Copper Rear View (FX3400-T)



Figure 8: Series 3 Fiber Rear View (FX3500-F)



Figure 9: Series 3 Copper Rear View (FX3500-T)

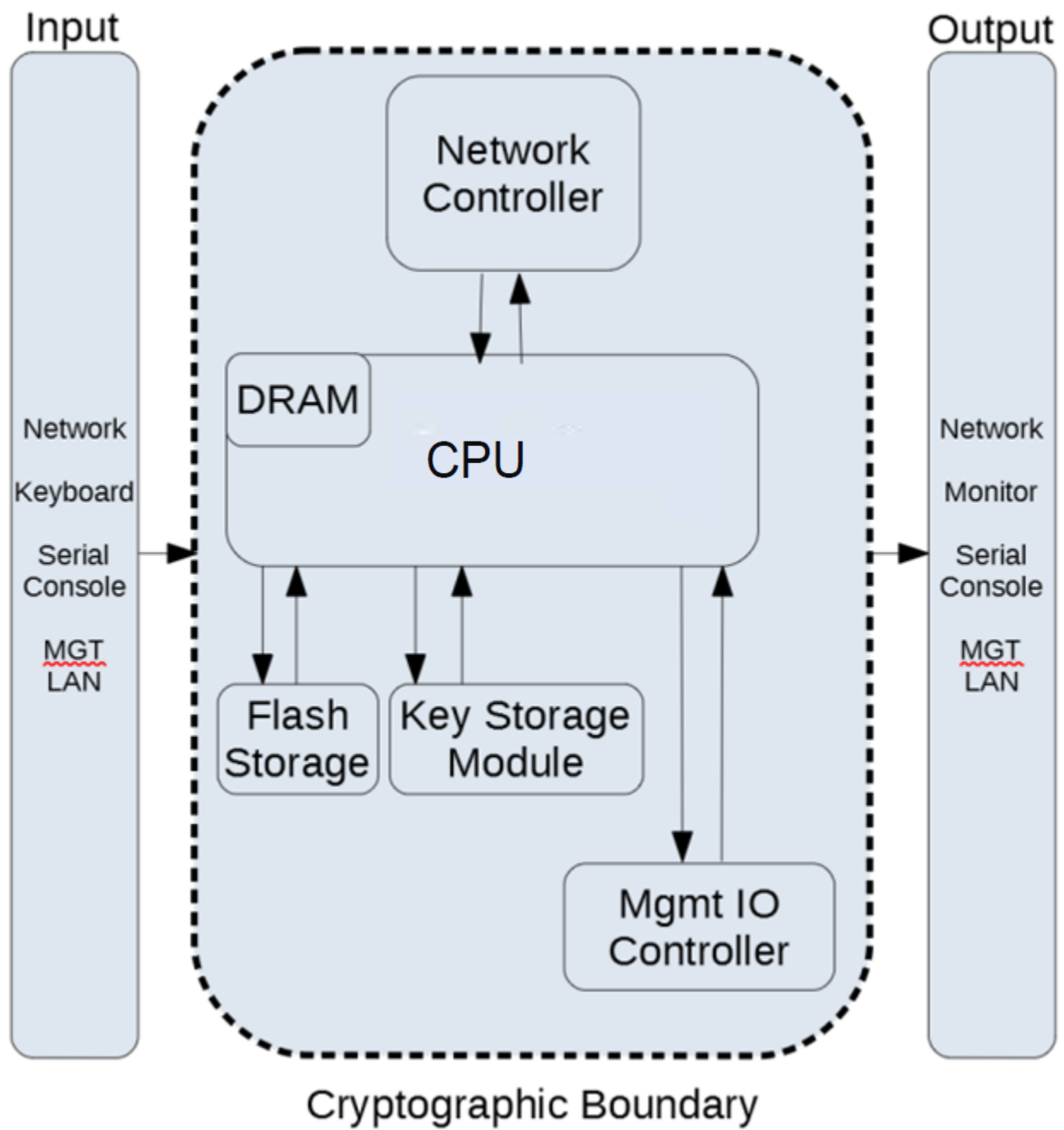


Figure 10: Hardware Block Diagram

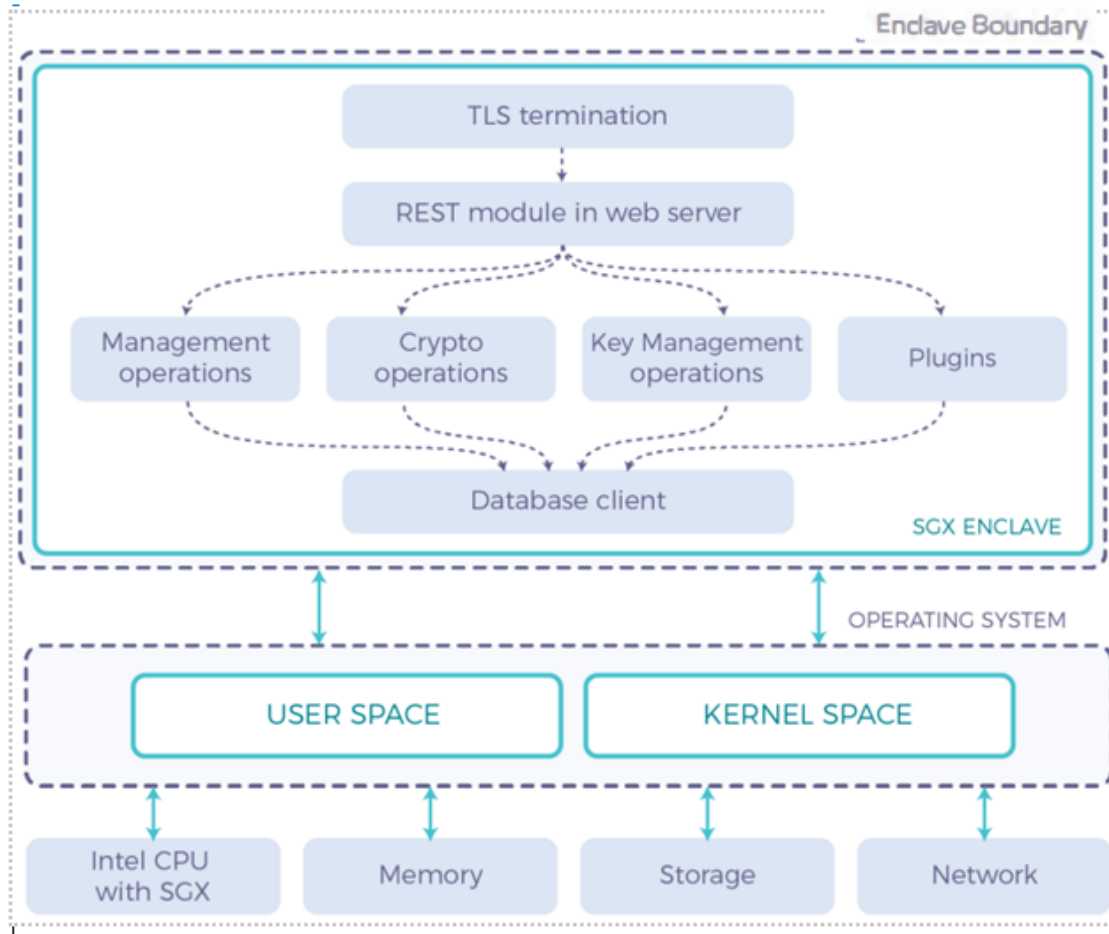


Figure 11: Firmware Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
FX2200-II-TN-F	FX2200-II-TN-F	4.28.2471 with plugin digest Xfbg4nYTWdMKgnUFjimfzAOBU0VF9Vz0PkGY P11MIFY=	Intel(R) Xeon(R) E-2176G CPU (Coffee Lake) @ 3.70GHz	The FX2200-II-TN-F unit is built with copper
FX2200-II-SXN-F	FX2200-II-SXN-F	4.28.2471 with plugin digest Xfbg4nYTWdMKgnUFjimfzAOBU0VF9Vz0PkGY P11MIFY=	Intel(R) Xeon(R) E-2176G CPU	FX2200-II-SXN-F unit is

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
			(Coffee Lake) @ 3.70GHz	built with fiber
FX3400-T	FX3400-T	4.28.2471 with plugin digest Xfbg4nYTWdMKgnUFjimfzAOBU0VF9Vz0PkGY P11MIFY=	Intel(R) Xeon(R) Silver 4410T CPU (Sapphire Rapids) @ 2.00 GHz	The FX3400-T unit has copper connectivity
FX3400-F	FX3400-F	4.28.2471 with plugin digest Xfbg4nYTWdMKgnUFjimfzAOBU0VF9Vz0PkGY P11MIFY=	Intel(R) Xeon(R) Silver 4410T CPU (Sapphire Rapids) @ 2.00 GHz	FX3400-F unit has both copper and fiber
FX3500-T	FX3500-T	4.28.2471 with plugin digest Xfbg4nYTWdMKgnUFjimfzAOBU0VF9Vz0PkGY P11MIFY=	Intel(R) Xeon(R) Gold 6421N CPU (Sapphire Rapids) @ 1.80GHz	The FX3500-T unit has copper connectivity
FX3500-F	FX3500-F	4.28.2471 with plugin digest Xfbg4nYTWdMKgnUFjimfzAOBU0VF9Vz0PkGY P11MIFY=	Intel(R) Xeon(R) Gold 6421N CPU (Sapphire Rapids) @ 1.80GHz	FX3500-F unit has both copper and fiber

Table 3: Tested Module Identification – Hardware

2.3 Excluded Components

The following non-security relevant components have been excluded from FIPS 140-3 requirements:

- Power distributor (R1T2-5300G2H, R1T2-5500K2H) board / circuitry for power supply receptacle (non-security relevant)
- Wiring connected to Front bezel LEDs and buttons (non-security relevant)
- Components U2 (NOR Gate) and C6 (Capacitor) casing on network card (PE3251G21/1-SR) (non-security relevant)
- Component U42 (32-bit CMOS Flash) on main board (MX32-4L0) (non-security relevant)

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved Mode	Invoke the "system version" CLI command provided by the "Get status" service. Verify that the versions match with the validated version, and the value of the "fips_level" attribute is 3	Approved	API Version: "v1-20170718" Server Mode: "Sgx" FIPS Level: 3 Plugin Digest: "Xfbg4nYTWdMKgnUFjimfzAOBU0VF9Vz0PkGYP11MIFY=" Hardware Name: Fortanix DSM Appliance Hardware Version: [FX2200-II-T/F, FX3400-T/F, FX3500-T/F] Firmware Name: Data Security Manager Firmware Version: 4.28.2471

Table 4: Modes List and Description

The module always operates in the approved mode. The Crypto Officer shall follow the steps in Section 11 for initialization and installation and in the preceding table to verify the module is running in the Approved Mode.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5344	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A5344	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A5344	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A5344	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A5344	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A5244	Direction - Encrypt Key Length - 128	SP 800-38A
AES-ECB	A5245	Direction - Encrypt Key Length - 128	SP 800-38A
AES-ECB	A5344	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-FF1	A5344	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38G
AES-GCM	A5344	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
AES-KW	A5344	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KWP	A5344	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A5344	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
Conditioning Component AES-CBC- MAC SP800-90B	A5244	Key Length - 128	SP 800-90B
Conditioning Component AES-CBC- MAC SP800-90B	A5245	Key Length - 128	SP 800-90B
Counter DRBG	A5344	Prediction Resistance - No, Yes Mode - AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A5344	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A5344	Curve - P-224, P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A5344	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3- 256, SHA3-384, SHA3-512 Component - No, Yes	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A5344	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256,	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
		SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	
EDDSA KeyGen	A5344	Curve - ED-25519	FIPS 186-5
EDDSA SigGen	A5344	Curve - ED-25519 PreHash - No, Yes Pure - Yes	FIPS 186-5
EDDSA SigVer	A5344	Curve - ED-25519 PreHash - No Pure - Yes	FIPS 186-5
HMAC-SHA-1	A5344	Key Length - Key Length: 112-8192 Increment 8	FIPS 198-1
HMAC-SHA2-224	A5344	Key Length - Key Length: 112-8192 Increment 8	FIPS 198-1
HMAC-SHA2-256	A5344	Key Length - Key Length: 112-8192 Increment 8	FIPS 198-1
HMAC-SHA2-384	A5344	Key Length - Key Length: 112-8192 Increment 8	FIPS 198-1
HMAC-SHA2-512	A5344	Key Length - Key Length: 112-8192 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A5344	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder onePassDh - KAS Role - initiator, responder staticUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A5344	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 Scheme - dhEphem - KAS Role - initiator, responder dhOneFlow - KAS Role - initiator, responder dhStatic - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDF SP800-108	A5344	KDF Mode - Feedback Supported Lengths - Supported Lengths: 256	SP 800-108 Rev. 1
KTS-IFC	A5344	Modulo - 2048, 3072, 4096, 6144, 8192 Key Generation Methods - rsakpg1-basic Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Key Length - 1024	SP 800-56B Rev. 2

Algorithm	CAVP Cert	Properties	Reference
LMS KeyGen	A5344	-	SP 800-208
LMS SigGen	A5344	-	SP 800-208
LMS SigVer	A5344	-	SP 800-208
PBKDF	A5344	Iteration Count - Iteration Count: 100000 Password Length - Password Length: 8-128 Increment 8	SP 800-132
RSA Decryption Primitive Sp800-56Br2 (CVL)	A5344	Modulo - 2048, 3072, 4096	SP 800-56B Rev. 2
RSA KeyGen (FIPS186-5)	A5344	Key Generation Mode - probable Modulo - 2048, 3072, 4096, 8192 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A5344	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
RSA Signature Primitive (CVL)	A5344	Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-2)	A5344	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-5)	A5344	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
Safe Primes Key Generation	A5344	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	SP 800-56A Rev. 3
Safe Primes Key Verification	A5344	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	SP 800-56A Rev. 3
SHA-1	A5344	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-224	A5344	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A5344	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A5345	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A5344	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A5344	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA3-224	A5344	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-256	A5344	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-384	A5344	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-512	A5344	Message Length - Message Length: 0-65536 Increment 8	FIPS 202

Algorithm	CAVP Cert	Properties	Reference
SHAKE-128	A5344	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A5344	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
TLS v1.2 KDF RFC7627 (CVL)	A5344	Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A5344	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE	SP 800-135 Rev. 1

Table 5: Approved Algorithms

There are algorithm modes that were validated but not implemented by the module. Only the algorithms, modes, and key sizes that are implemented by the module are shown in this table.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type:Symmetric and Asymmetric	N/A	NIST SP800-133rev2 Section 4 Scenario 1

Table 6: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

The module does not implement any algorithms that fall under this categorization.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

The module does not implement any algorithms that fall under this categorization.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

The module does not implement any algorithms that fall under this categorization.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
AES-Auth	BC-Auth	AES Block Cipher in Authenticated Modes		AES-CCM: (A5344) Key Length: 128, 192, 256 Tag Length: 32, 48, 64, 80, 96,

Name	Type	Description	Properties	Algorithms
				112, 128 IV Length: 56, 64, 72, 80, 88, 96, 104 Payload Length: 0-256 Increment 8 AAD Length: 0- 256 Increment 8 AES-GCM: (A5344) Direction: Decrypt, Encrypt IV Generation: Internal IV Generation Mode: 8.2.1 Key Length: 128, 192, 256 Tag Length: 32, 64, 96, 104, 112, 120, 128 IV Length: 8-128 Increment 8 Payload Length: 0-65536 Increment 8 AAD Length: 0- 65536 Increment 8 AES-KW: (A5344) Direction: Decrypt, Encrypt Cipher: Cipher Key Length: 128, 192, 256 Payload Length: 128-4096 Increment 64 AES-KWP: (A5344) Direction: Decrypt, Encrypt Cipher: Cipher Key Length: 128, 192, 256 Payload Length: 8-4096 Increment 8

Name	Type	Description	Properties	Algorithms
AES-UnAuth	BC-UnAuth	AES Block Cipher in Unauthenticated Modes		AES-CBC: (A5344) AES-CFB128: (A5344) AES-CTR: (A5344) AES-FF1: (A5344) AES-ECB: (A5344) AES-OFB: (A5344)
AES-KTS-Wrap-API	KTS-Unwrap KTS-Wrap	AES Key Transport via API, wrap and unwrap	Standard :SP 800-38F IG D.G:Approved Caveat:Key establishment provides between 128 and 256 bits of security strength	AES-CCM: (A5344) AES-GCM: (A5344) AES-KW: (A5344) AES-KWP: (A5344)
CMAC	MAC	CMAC Message Authentication		AES-CMAC: (A5344)
AES-KTS-Wrap-TLS	KTS-Unwrap KTS-Wrap	AES Key Transport via TLS	Standard :SP 800-38F IG D.G:Approved Caveat:Key establishment provides between 128 and 256 bits of security strength	AES-GCM: (A5344)
DRBG	CKG DRBG	Random number generator for SSP generation		Counter DRBG: (A5344) CKG: () Key Type: Symmetric and Asymmetric
ECDSA-AsymKeyPair-KeyGen	AsymKeyPair-KeyGen CKG	ECDSA Key-Pair Generation		ECDSA KeyGen (FIPS186-5): (A5344) CKG: () Key Type: Symmetric and Asymmetric

Name	Type	Description	Properties	Algorithms
ECDSA-AsymKeyPair-PubKeyVal	AsymKeyPair-PubKeyVal	ECDSA Public Key Validation		ECDSA KeyVer (FIPS186-5): (A5344)
ECDSA-DigSig-SigGen	DigSig-SigGen	ECDSA Signature Generation		ECDSA SigGen (FIPS186-5): (A5344)
ECDSA-DigSig-SigVer	DigSig-SigVer	ECDSA Signature Verification		ECDSA SigVer (FIPS186-5): (A5344)
EDDSA-AsymKeyPair-KeyGen	AsymKeyPair-KeyGen CKG	EDDSA Key-Pair Generation		EDDSA KeyGen: (A5344) CKG: () Key Type: Symmetric and Asymmetric
EDDSA-DigSig-SigGen	DigSig-SigGen	EDDSA Signature Generation		EDDSA SigGen: (A5344)
EDDSA-DigSig-SigVer	DigSig-SigVer	EDDSA Signature Verification		EDDSA SigVer: (A5344)
HMAC	MAC	HMAC Message Authentication		HMAC-SHA-1: (A5344) HMAC-SHA2-224: (A5344) HMAC-SHA2-256: (A5344) HMAC-SHA2-384: (A5344) HMAC-SHA2-512: (A5344)
KAS-ECC-SSC	KAS-SSC	ECC Key Agreement		KAS-ECC-SSC Sp800-56Ar3: (A5344)
KAS-FFC-SSC	KAS-SSC	FFC Key Agreement		KAS-FFC-SSC Sp800-56Ar3: (A5344)
KBKDF	KBKDF	Key-Based		KDF SP800-108: (A5344)
RSA-KTS-Wrap	KTS-Unwrap KTS-Wrap	Wrapping and Unwrapping	Standard:SP 800-56Brev2 IG D.G:Approved Key confirmation:no Caveat:Key establishment methodology	KTS-IFC: (A5344) RSA Decryption Primitive Sp800-56Br2: (A5344)

Name	Type	Description	Properties	Algorithms
			provides between 112 and 202 bits of security strength	
LMS-AsymKeyPair-KeyGen	AsymKeyPair-KeyGen CKG	LMS Key-Pair Generation		LMS KeyGen: (A5344) CKG: () Key Type: Symmetric and Asymmetric
LMS-DigSig-SigGen	DigSig-SigGen	LMS Signature Generation		LMS SigGen: (A5344)
LMS-DigSig-SigVer	DigSig-SigVer	LMS Signature Verification		LMS SigVer: (A5344)
PBKDF	PBKDF	Password-Based Key Derivation	SP 800-132:Option 1a with HMAC-SHA2-512	PBKDF: (A5344)
RSA-AsymKeyPair-KeyGen	AsymKeyPair-KeyGen CKG	RSA Key-Pair Generation		RSA KeyGen (FIPS186-5): (A5344) CKG: () Key Type: Symmetric and Asymmetric
RSA-DigSig-SigGen	DigSig-SigGen	RSA Signature Generation		RSA SigGen (FIPS186-5): (A5344) RSA Signature Primitive: (A5344)
RSA-DigSig-SigVer	DigSig-SigVer	RSA Signature Verification		RSA SigVer (FIPS186-5): (A5344)
SafePrimes-AsymKeyPair-KeyGen	AsymKeyPair-KeyGen CKG	Safe Primes Key-Pair Generation		Safe Primes Key Generation: (A5344) CKG: () Key Type: Symmetric and Asymmetric
SafePrimes-AsymKeyPair-KeyVer	AsymKeyPair-KeyVer	Safe Primes Key-Pair Verification		Safe Primes Key Verification: (A5344)
SHA	SHA	Secure Hash		SHA-1: (A5344) SHA2-224: (A5344) SHA2-256: (A5344)

Name	Type	Description	Properties	Algorithms
				SHA2-384: (A5344) SHA2-512: (A5344) SHA3-224: (A5344) SHA3-256: (A5344) SHA3-384: (A5344) SHA3-512: (A5344)
SHA-SIMD	SHA	Secure Hash		SHA2-256: (A5345)
XOF	XOF	Extendable Output Function		SHAKE-128: (A5344) SHAKE-256: (A5344)
TLS-KAS-135KDF	KAS-135KDF	SP800-135 Key Derivation		TLS v1.2 KDF RFC7627: (A5344) TLS v1.3 KDF: (A5344)
TLS-KAS-Full	KAS-Full	Full KAS	IG:IG D.F Scenario 2, path (2), split Key confirmation:No Key derivation:IG 2.4.B SP 800- 135rev1 CVL and TLS 1.3 KDF Caveat:Key establishment methodology provides between 112 and 256 bits of security strength	KAS-FFC-SSC Sp800-56Ar3: (A5344) KAS-ECC-SSC Sp800-56Ar3: (A5344) TLS v1.2 KDF RFC7627: (A5344) TLS v1.3 KDF: (A5344)
ENT-Cond	ENT-Cond	Conditioning Component		AES-ECB: (A5244, A5245) Conditioning Component AES-CBC-MAC SP800-90B: (A5244, A5245)
ENT-ESV	ENT-ESV	Entropy Source		

Name	Type	Description	Properties	Algorithms
RSA-DigSig-SigVer (Legacy)	DigSig-SigVer	RSA Signature Verification		RSA SigVer (FIPS186-2): (A5344)

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

Algorithms designated as “Legacy” can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M, September 2020 for 186-2 RSA Signature Verification.

The module implements the following algorithm-related restrictions:

- Module does not allow P-192 and/or SHA-1 for ECDSA signature generation. The minimum hash sizes allowed by the module are SHA2-256 for P-224, SHA2-256 for P-256, SHA2-384 for P-384, and SHA2-512 for P-521. Module does not allow key pair generation with P-192.
- Module does not allow 1024-bit keys and/or SHA-1 for RSA Signature Generation.
- SHA-1 is non-approved for digital signature generation and deprecated through December 31, 2030 for non-digital signature applications.
- SHA-1 is only used for legacy signature verification only.

The module complies with FIPS 140-3 IG C.H requirements for AES-GCM:

- For TLS V1.2 Protocol, the module constructs the IV (internally) as allowed per Technique #1 in FIPS 140-3 IG C.H for Industry Protocols. The AES-GCM implementation complies with RFC 5288 and SP 800-52. The IV total length is 96-bits, where the fixed IV length is 32-bits and nonce_explicit part of the IV is a 64-bit counter. The GCM key and IV are session specific; if the module loses power the implementation re-initializes a TLS V1.2 session, creating a new IV altogether. The implementation ensures that when the counter exceeds the maximum value, the session is renegotiated, and new keys are established.
- For TLS V1.3 Protocol, the module constructs the IV (internally) as allowed per Technique #5 in FIPS 140-3 IG C.H for Industry Protocols. The IV total length is 96-bits, where a 64-bit record sequence number is left-padded and XORed with a mutually derived IV (as specified in RFC 8446). The mutually derived IV is derived from a secret that is originally based on 32 bytes of random data generated by the module. This assures that the odds of a collision are less than $2^{(-32)}$. The GCM key and IV are session specific; if the module loses power the implementation is required to re-initialize a TLS V1.3 session, creating a new IV altogether.
- For the Encrypt/Decrypt service, a 96-bit IV is constructed from the output of the CTR_DRBG, allowed as per Technique #2 in FIPS 140-3 IG C.H for IVs generated “internally at its entirety randomly”. In case the module’s power is lost and then restored, a new IV for use with the AES GCM encryption/decryption will be generated from the output of the CTR_DRBG.
- The module complies with IG D.N requirements for PBKDF. The module supports option 1a with HMAC SHA2-512. The module supports an iteration count of 100,000 which is much larger than the required minimum of 1000. For the justification for the password

length chosen for PBKDF please see Section 4.1 below. Keys derived from the PBKDF may only be used in storage applications.

2.8 RBG and Entropy

Cert Number	Vendor Name
E232	Fortanix
E233	Fortanix

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Fortanix DRNG RDSEED Entropy Source (Coffee Lake 6 Core) within FX2200-II-TN-F, FX2200-II-SXN-F	Physical	Intel Corporation Intel(R) Pentium(R), Core(R) and Xeon(R) Coffee Lake 6 Core Die with FCLGA1151 Package	128	128	A5244
Fortanix DRNG RDSEED Entropy Source (Sapphire Rapids-MCC) within FX3400-T, FX3400-F, FX3500-T, FX3500-F	Physical	Intel Corporation 4th Gen Intel(R) Xeon(R) Processor Scalable Family (Sapphire Rapids-MCC) Core Die with FCLGA4677 Package	128	128	A5245

Table 9: Entropy Sources

The module's DRBG is seeded with a minimum of 384 bits of entropy which is sufficient to generate keys with up to 256 bits of security strength.

2.9 Key Generation

The module supports the ability to generate symmetric (AES, HMAC) and asymmetric keys (EC, Diffie-Hellman, LMS, RSA) using the direct output of the DRBG. The module supports CKG using the following SP 800-133Rev2 sections:

- SP 800-133Rev2 Section 5.1 Asymmetric signature key generation using unmodified DRBG output
- 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 6.2.2 Derivation of symmetric keys from a pre-shared key

2.10 Key Establishment

The module supports key establishment via (EC) Diffie-Hellman key agreement, AES and RSA-based key transport, key generation (see section 2.9 Key Generation), SSP entry via a trusted channel (see section 3.2 Trusted Channel Specification), key derivation, as well as pre-loading of a key. See section 2.6 Security Function Implementations for the list of implementations used for key establishment, as well as section 9.4 SSPs for the list of keys that are pre-loaded.

2.11 Industry Protocols

- The module supports the TLS 1.2 and TLS 1.3 protocols as defined in RFC 5246 and RFC 8446, respectively.
- All API calls into the module are done over TLS V1.2 or TLS V1.3. Per IG D.C, no parts of these protocols, other than the KDFs, have been validated by the CAVP and CMVP.
- Per IG D.Q, the extended master secret is used.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
Networking Ports	Data Input Data Output Control Input Status Output	Encrypted Input / Request payload of API, which can include SSPs and data; Encrypted Output / Response payload of API, which can include SSPs and data; Encrypted API calls; Encrypted API status information and return status codes provided by API
IPMI Port	Data Input Data Output Control Input Status Output	Plaintext Input / Request payload of CLI, which can include SSPs; Plaintext Output / Response payload of CLI, which can include PSPs; CLI calls; CLI returning status information and return status codes provided by CLI This port is primarily used for Chassis management
VGA Port	Data Output Status Output	Plaintext Output / Response payload of CLI, which can include PSPs; CLI returning status information and return status codes provided by CLI
Serial Port	Data Input Data Output Control Input Status Output	Plaintext Input / Request payload of CLI, which can include SSPs; Plaintext Output / Response payload of CLI, which can include PSPs; CLI calls; CLI returning status information and return status codes provided by CLI
USB Ports	Data Input Control Input Power	Plaintext Input / Request payload of CLI via Keyboard, which can include SSPs; CLI calls via Keyboard; Power Output for connected devices

Physical Port	Logical Interface(s)	Data That Passes
Power Receptacle	Power	Power
Network Link LEDs	Status Output	Status output via LEDs (Green); Blinking: Indicates that link is up and there is link activity; Steady On: Indicates that link is up but there is no link activity; Off: Powered off when the unit is off
Hard Disk Activity LED	Status Output	Status output via LEDs (Amber); Off: Indicates the system is not on; Blinking amber: Indicates there is disk activity
Power Supply Failure Indicator LED	Status Output	Status output via LEDs (Blue); Off: Indicates the system is not on; Blue: Indicates the system is healthy and operating fine
Power Button	Control Input	Manual Control for Power
ID Button	Control Input	Manual Control for Self-Identification
Reset Button	Control Input	Manual Control for Reset
Front panel display LED	Status Output	Self-Identification

Table 10: Ports and Interfaces

The table above describes the physical ports and logical interfaces of the module. The logical interfaces are implemented as application programming interfaces (API) and the command-line interface (CLI). The logical interfaces of the module offer services. The applications interacting with the module input control information and data to the module through the input fields of the API and CLI and receive output data and/or status information via the output of the API and CLI. API and CLI documentation describe in detail the successful operation output and error in case of a failed operation.

At the API level, logical separation is maintained through the usage of labels within the JSON-based requests and responses. Based on the specific label for a particular value, the module can classify that value as data, control, or status information. At the CLI level, logical separation is maintained through the specific command specifications as well as the usage of positional arguments that are passed to the commands. Based on the specific command that is being executed along with where an argument is positioned in a command, the module can determine whether it is data, control, or status information.

3.2 Trusted Channel Specification

The module's CLI is a trusted channel, which protects plaintext CSPs through physical means. To set up and maintain the trusted channel, the operator must establish a direct connection to the module's serial port/IPMI with a non-networked general-purpose computer (GPC) using an RJ45 cable. The operator must stay in control of the non-networked GPC, RJ45 cable, and the module to prevent any unauthorized tampering (modification, substitution, or disclosure) along the communication link. The module's serial port is used only for the purpose of the trusted channel. The usage of the module's trusted channel is indicated at the CLI via the banner:

```

*****
* Welcome to DSM Trusted Channel Interface *
*
*      WARNING: Authorised Access Only      *
*****

```

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Username and Password	Memorized Secret	Username and Password	The minimum password length is 8 characters. For a user who just meets the minimum password length, each of the eight characters will have at least 95 possible characters if we consider just the printable ASCII characters, although module supports UTF-8 characters for password and the number of possible characters with UTF-8 is much higher. Total number of password permutations with eight characters is $95^8 = 6,634,204,312,890,625$. Therefore, the probability of guessing a password with a single random guess is significantly less than one in 1,000,000.	The module only allows at the most 100 failed password-based authentication attempts in a minute. Given the total number of possible permutations (as shown above), the probability of a random attempt in one-minute period to be correct will be $100 / 6,634,204,312,890,625$. Therefore, the probability of guessing a password in a one-minute period is significantly less than one in 100,000.
API Key	Memorized Secret	DRBG is used to generate the API keys. Authentication is done over TLS (AES-KTS-Wrap-TLS)	An application authenticates using an API key which contains application ID and application secret. App secret is 16 - 64 bytes random data. Total number of permutations for app	The module only allows at the most 100 authentication attempts in a minute. Given the total number of possible permutations (as shown above), the probability of a random

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
			secret will be 2^{128} in the worst case. Therefore, the probability of guessing an application's secret is significantly less than one in 1,000,000.	attempt in one-minute period to be correct will be $100 / (2^{128})$. Therefore, the probability of guessing an app secret in a one-minute period is significantly less than one in 100,000.
User 2FA device public key	Multi-Factor Crypto Firmware	ECDSA-DigSig-SigVer (P-256) is used to verify signatures	The module allows users to use a second factor authentication mechanism in addition to username and password. The strength of this combination mechanism relies upon the strength of the User password mechanism (described earlier) combined with the strength of two factor authentication. This mechanism adds more strength to the password mechanism which already far exceeds the FIPS 140-3 requirements. U2F signature verification uses U2F device's public key which is an EC P-256 key. The security strength of this key is 128 bits. So, the probability of a random success will be 1 in 2^{128} . Probability of this combined scheme = (Probability of guessing username and password) * (Probability from signature verification scheme), which is $1 / (95^8) * 1 / (2^{128})$. Therefore, the probability of guessing	The module only allows at the most 100 authentication attempts in a minute. Given the total number of possible permutations (as shown above), the probability of a random attempt in one-minute period to be correct will be $100 / (95^8 * 2^{128})$. Therefore, the probability of guessing a valid password and 2FA signature in a one-minute period is significantly less than one in 100,000. Therefore, this mechanism of additional 2FA also far exceeds the FIPS 140-3 requirements.

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
			a password is significantly less than one in 1,000,000.	
RSA Public key of external Application	Single-Factor Crypto Firmware	RSA-DigSig-SigVer is used to verify signatures	The strength of this mechanism is based on the size of the private key space. The module relies upon minimum RSA 2048-bit keys. This provides an encryption strength of 112 bits, so the probability of a random success will be 1 in 2^{112} , which is significantly less than one in 1,000,000.	Using this mechanism, one can make very few attempts in a one-minute period. Each attempt will require the module to check the signature on the certificate using approved signature algorithm and establishing TLS session with this certificate. On average only one attempt can be made in a second. Therefore, at the most 60 attempts can be made in a one-minute period. Therefore, the probability of guessing a 2048-bit private key and succeeding in a one-minute period is $60 / (2^{112})$ which is significantly less than one in 100,000.
Bearer token	Memorized Secret	DRBG is used to generate the Bearer token. Authentication is done over TLS (AES-KTS-Wrap-TLS)	This authentication mechanism builds upon other authentication mechanisms, and it maps to the original authentication credentials that were used to establish an authenticated session. The bearer token is a base64 encoded random 64 bytes data which is generated using approved DRBG in DSM. Total number of permutations is 2^{512} . Therefore, the probability of guessing the token is 1 /	Each authentication attempt takes approximately 12ms or more. Therefore, a user could try at most 5,000 attempts in a minute. Given the total number of possible permutations (as shown above), the probability a random attempt in one-minute period to be correct will be $5000 / (2^{512})$. Therefore, the probability of guessing a password in a one-minute period is significantly less than one in 100,000.

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
			(2^{512}), which is significantly less than one in 1,000,000.	
Public key of an outside entity/server (CA)	Single-Factor Crypto Firmware	RSA-DigSig-SigVer is used to verify signatures	The strength of this mechanism is based on the size of the private key space. The module relies upon RSA 2048-bit node keys. This provides an encryption strength of 112 bits, so the probability of a random success will be 1 in 2^{112} , which is significantly less than one in 1,000,000.	Each attempt will require the module to check the signature on the certificate using approved signature algorithm and establishing TLS session with this certificate. Each attempt takes 100ms or more. Therefore, at the most 600 attempts can be made in a one-minute period. Therefore, the probability of guessing a 2048-bit private key and succeeding in a one-minute period is $600 / (2^{112})$ which is significantly less than one in 100,000.
JWT Public key of external application	Single-Factor Crypto Firmware	ECDSA-DigSig-SigVer or RSA-DigSig-SigVer is used to verify signatures	The strength of this mechanism is based on the size of the private key space. The module relies upon minimum RSA 2048-bit keys. This provides an encryption strength of 112 bits, so the probability of a random success will be 1 in 2^{112} , which is significantly less than one in 1,000,000.	Each attempt will require the module to check the signature on the token using approved signature algorithm and establishing the associated TLS session with the client. Each attempt takes 100ms or more. Therefore, at the most 600 attempts can be made in a one-minute period. Therefore, the probability of guessing a 2048-bit private key and succeeding in a one-minute period is $600 / (2^{112})$ which is significantly less than one in 100,000.

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
AWS Signature Version 4	Single-Factor Crypto Firmware	HMAC is used to verify authentication	The strength of this mechanism is based on the size of the secret key space. The module relies upon 32-byte (256-bit) HMAC keys. This provides security strength of 256 bits, so the probability of a random success will be 1 in 2^{256} , which is significantly less than one in 1,000,000.	Each attempt will require the module to check the authentication code on the request using approved HMAC algorithm and establishing the associated TLS session with the client. Each attempt takes 100ms or more. Therefore, at the most 600 attempts can be made in a one-minute period. Therefore, the probability of guessing a 32-byte secret key and succeeding in a one-minute period is $600 / (2^{256})$ which is significantly less than one in 100,000.

Table 11: Authentication Methods

The module supports identity-based authentication for all operators. The module supports a Crypto Officer and User Role.

- The Crypto Officer installs and administers the module.
- The User uses the cryptographic services provided by the module. This role is assumed both by an actual user of the system and an external system that requires cryptographic services.

Role separation is maintained by associating a specific login session with the specific port being utilized in the case of CLI authentication and using uniquely generated bearer tokens in the case of Web GUI and API authentications. When the module is power cycled or rebooted, the user is required to re-authenticate with the module. In the case of already-authenticated sessions, this can involve the operator re-presenting the bearer token to the module via the Web GUI or API request headers.

The module's password authentication policy is as described for the Memorized Secret Authenticators in NIST SP 800-63B (8 characters or longer). The module supports concurrent operators, and the module levies a restriction on session expiry time where if inactive, the Application's role session will expire in 10 minutes by default. Similarly, for all other module roles there is a session expiry time of 24 hours. Session expiry time can be customized.

A failed login attempt will result in the message "Login Failed. Incorrect username or password"

4.2 Roles

System Administrator (SA) A user with this role is responsible for installing, configuring, upgrading, and managing DSM. Key management and cryptographic operations are not available for SA.

System Operator (SO) A user with this role is responsible for monitoring and troubleshooting DSM. Key management and cryptographic operations are not available for SO.
Account Administrator (AA) A user with this role can create an account, add groups, users, and applications to the account. Key management and limited cryptographic operations are available for AA.

Account Member (AM) An Account Member is an administrator or auditor for one or more groups in an account. Key management and limited cryptographic operations are available for AA. As a group administrator, an account member may create applications and security objects in the group and can invite other users to the group. As a group auditor, an account member may only view objects and their associated audit logs in the group.

Account Auditor (AAU) A user with this role can view various entities in the account and their associated audit logs but can't perform any action. Key management and cryptographic operations are not available for AAU.

Group Administrator (GA) A user with this role can work in the specific groups that they are entitled to. In those groups the user can create add / modify users, applications and security objects. Key management and limited cryptographic operations are available for GA.

Group Auditor (GAU) A user with this role can view various entities in the specific groups that they are entitled to, and their associated audit logs, but can't perform any action. Key management and cryptographic operations are not available for GAU.

Applications (App) An application can programmatically access DSM and can perform cryptographic operations in addition to key management operations. Key management and cryptographic operations are available for App.

Console User (CU) This is the user who logs in using the appliance console to perform appliance setup and maintenance.

Name	Type	Operator Type	Authentication Methods
System Administrator	Identity	Crypto Officer	Username and Password User 2FA device public key Bearer token
System Operator	Identity	Crypto Officer	Username and Password User 2FA device public key Bearer token
Account Administrator	Identity	Crypto Officer, User	Username and Password User 2FA device public

Name	Type	Operator Type	Authentication Methods
			key Bearer token
Account Member	Identity	Crypto Officer, User	Username and Password User 2FA device public key Bearer token
Account Auditor	Identity	Crypto Officer	Username and Password User 2FA device public key Bearer token
Group Administrator	Identity	Crypto Officer, User	Username and Password User 2FA device public key Bearer token
Group Auditor	Identity	Crypto Officer	Username and Password User 2FA device public key Bearer token
Application	Identity	User	API Key RSA Public key of external Application Bearer token Public key of an outside entity/server (CA) JWT Public key of external application AWS Signature Version 4
Console User	Identity	Crypto Officer	Username and Password

Table 12: Roles

The module supports a variety of roles that are mapped to the two FIPS 140-3 defined roles. The preceding table enumerates the mapping between the module roles and FIPS 140-3 defined roles. In addition to the built-in roles, the module also does support the ability to create custom roles which would have a subset of the permissions that are available to the built-in roles.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Zeroization (Perform Zeroisation)	Zeroization command to wipe all SSPs	[IND_1] The module's CLI provides indicators in the form of	CLI command	CLI status	None	Console User - Personalization Key: Z - Firmware update key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		human-readable outputs, with success indicators being output to the 'stdout' output stream and error indicators being output to the 'stderr' output stream				- Plugin signing key: Z - Persisted Sealing Key: Z - Cluster Master Key: Z - System key: Z - Account Wrapping key: Z - Account key: Z - Database Wrapping key: Z - Cipher State Wrapping key: Z - Symmetric key: Z - HMAC key: Z - RSA private key for Digital Signatures: Z - RSA private key for Key Encapsulation Operations: Z - RSA public key for Key Encapsulation Operations: Z - Cluster RSA private key for TLS: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- ECDSA random number "k": Z - Cluster RSA public key for TLS: Z - LMS seed: Z - TLS KDF internal state: Z - TLS integrity key: Z - TLS encryption key: Z - TLS pre-master secret: Z - TLS extended master secret: Z - SP 800-108 KDF internal state: Z - User password : Z - API key: Z - AWS Secret Access Key: Z - Bearer token: Z - Public key of an outside entity/server (CA): Z - RSA Public key of external Application:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Z - User 2FA device public key: Z - JWT Public key of external Application / CA: Z - RSA public key for Digital Signatures: Z - User Key: Z - Console User Key: Z - User Salt: Z - Console User Salt: Z - CTR_DRBG Seed: Z - CTR_DRBG entropy input: Z - CTR_DRBG V: Z - CTR_DRBG Key: Z - ECDSA private key: Z - ECDSA Public key: Z - EDDSA private key: Z - EDDSA public key: Z - LMS private key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- LMS public key: Z - ECCDH private key : Z - ECCDH public key: Z - FFCDH private key: Z - FFCDH public key: Z
TLS (Perform approved security functions)	TLS for REST API and Web GUI	[IND_2] The module's REST API and Web GUI provide indicators in the form of HTTP status codes. An HTTP status code of 2xx is indicative of success, while 4xx and 5xx status codes indicate errors. The HTTP status codes are also delivered alongside human-readable outputs. Full	API and GUI requests	API and GUI responses	AES-Auth DRBG ECDSA-AsymKeyPair-KeyGen ECDSA-AsymKeyPair-PubKeyVal ECDSA-DigSig-SigGen ECDSA-DigSig-SigVer HMAC KAS-ECC-SSC KAS-FFC-SSC RSA-DigSig-SigGen RSA-DigSig-SigVer SafePrimes - AsymKeyPair-KeyGen SafePrimes - AsymKeyPair-KeyVer SHA TLS-KAS-135KDF	System Administrator - Cluster RSA private key for TLS: R,E - TLS KDF internal state: G,E,Z - TLS integrity key: G,E,Z - TLS encryption key: G,E,Z - TLS pre-master secret: G,E,Z - TLS extended master secret: G,E,Z - ECCDH private key : G,E,Z - ECCDH public key: G,E,Z - - CTR_DRBG entropy input: G,R,W,E - Public key

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		details of this indicator can be found in the Fortanix DSM REST API specification [API]			TLS-KAS-Full ENT-Cond ENT-ESV RSA-DigSig-SigVer (Legacy)	of an outside entity/server (CA): R,E - RSA Public key of external Application: R,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E - Cluster RSA public key for TLS: R,E - FFCDH private key: G,E,Z - FFCDH public key: G,E,Z System Operator - Cluster RSA private key for TLS: R,E - TLS KDF internal state: G,E,Z - TLS integrity key: G,E,Z - TLS encryption key: G,E,Z - TLS pre-master secret: G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS extended master secret: G,E,Z - ECCDH private key : G,E,Z - ECCDH public key: G,E,Z - CTR_DRBG entropy input: G,R,W,E - Public key of an outside entity/server (CA): R,E - RSA Public key of external Application: R,E - Cluster RSA public key for TLS: R,E - FFCDH private key: G,E,Z - FFCDH public key: G,E,Z - Account Administrator - Cluster RSA private key for TLS: R,E - TLS KDF internal state: G,E,Z - TLS integrity key: G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS encryption key: G,E,Z - TLS pre-master secret: G,E,Z - TLS extended master secret: G,E,Z - ECCDH private key : G,E,Z - ECCDH public key: G,E,Z - CTR_DRBG entropy input: G,R,W,E - Public key of an outside entity/server (CA): R,E - RSA Public key of external Application: R,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E - Cluster RSA public key for TLS: R,E - FFCDH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						private key: G,E,Z - FFCDH public key: G,E,Z Account Member - Cluster RSA private key for TLS: R,E - TLS KDF internal state: G,E,Z - TLS integrity key: G,E,Z - TLS encryption key: G,E,Z - TLS pre- master secret: G,E,Z - TLS extended master secret: G,E,Z - ECCDH private key : G,E,Z - ECCDH public key: G,E,Z - CTR_DRBG entropy input: G,R,W,E - Public key of an outside entity/server (CA): R,E - RSA Public key of external Application:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						R,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E - Cluster RSA public key for TLS: R,E - FFCDH private key: G,E,Z - FFCDH public key: G,E,Z Account Auditor - Cluster RSA private key for TLS: R,E - TLS KDF internal state: G,E,Z - TLS integrity key: G,E,Z - TLS encryption key: G,E,Z - TLS pre- master secret: G,E,Z - TLS extended master secret: G,E,Z - ECCDH private key : G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- ECCDH public key: G,E,Z - CTR_DRBG entropy input: G,R,W,E - Public key of an outside entity/server (CA): R,E - RSA Public key of external Application: R,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E - Cluster RSA public key for TLS: R,E - FFCDH private key: G,E,Z - FFCDH public key: G,E,Z - Group Administrator - Cluster RSA private key for TLS: R,E - TLS KDF internal state: G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS integrity key: G,E,Z - TLS encryption key: G,E,Z - TLS pre-master secret: G,E,Z - TLS extended master secret: G,E,Z - ECCDH private key : G,E,Z - ECCDH public key: G,E,Z - CTR_DRBG entropy input: G,R,W,E - Public key of an outside entity/server (CA): R,E - RSA Public key of external Application: R,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E - Cluster RSA public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key for TLS: R,E - FFCDH private key: G,E,Z - FFCDH public key: G,E,Z Group Auditor - Cluster RSA private key for TLS: R,E - TLS KDF internal state: G,E,Z - TLS integrity key: G,E,Z - TLS encryption key: G,E,Z - TLS pre- master secret: G,E,Z - TLS extended master secret: G,E,Z - ECCDH private key : G,E,Z - ECCDH public key: G,E,Z - CTR_DRBG entropy input: G,R,W,E - Public key of an outside entity/server (CA): R,E - RSA Public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key of external Application: R,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E - Cluster RSA public key for TLS: R,E - FFCDH private key: G,E,Z - FFCDH public key: G,E,Z Application - Cluster RSA private key for TLS: R,E - TLS KDF internal state: G,E,Z - TLS integrity key: G,E,Z - TLS encryption key: G,E,Z - TLS pre-master secret: G,E,Z - TLS extended master secret: G,E,Z - ECCDH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						private key : G,E,Z - ECCDH public key: G,E,Z - CTR_DRBG entropy input: G,R,W,E - Public key of an outside entity/server (CA): R,E - RSA Public key of external Application: R,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E - Cluster RSA public key for TLS: R,E - FFCDH private key: G,E,Z - FFCDH public key: G,E,Z
Firmware update (Perform approved security functions)	Update the module firmware	IND_2	New firmware image (via API)	API response	ECDSA-DigSig-SigVer SHA	System Administrator - Firmware update key: R,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Plugin load (Perform approved security functions)	Load certified plugins	IND_2	Signed plugin (via API)	API response	ECDSA-DigSig-SigVer RSA-DigSig-SigVer SHA RSA-DigSig-SigVer (Legacy)	Account Administrator - Plugin signing key: R,E Account Member - Plugin signing key: R,E
Module Configuration (Perform approved security functions)	Initialization and configuration of module including setup of network, IPMI, DNS and NTP. Account management including users, groups, security objects and apps	IND_1, IND_2	CLI commands, API and GUI requests	CLI, API, and GUI responses	AES- UnAuth DRBG HMAC PBKDF RSA- AsymKeyPair-KeyGen SHA ENT-Cond ENT-ESV	System Administrator - User password : W - User 2FA device public key: W - Cluster Master Key: R,E - System key: G,R,W,E - SP 800-108 KDF internal state: R,W,E,Z - User Key: W - User Salt: W System Operator - User password : W - User 2FA device public key: W - Cluster Master Key: R,E - System

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key: G,R,W,E - SP 800-108 KDF internal state: R,W,E,Z - User Key: W - User Salt: W Account Administrator - User password : W - Plugin signing key: W - Public key of an outside entity/server (CA): W - API key: G,R - User 2FA device public key: W - JWT Public key of external Application / CA: W - AWS Secret Access Key: G,W - Cluster Master Key: R,E - System key: G,R,W,E - Account Wrapping key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,R,W,E - Account key: G,R,W,E - Database Wrapping key: G,R,W,E - SP 800-108 KDF internal state: R,W,E,Z - User Key: W - User Salt: W Account Member - User password : W - User 2FA device public key: W - Cluster Master Key: R,E - System key: G,R,W,E - SP 800-108 KDF internal state: R,W,E,Z - Public key of an outside entity/server (CA): W - API key: G,R - JWT Public key of external Application / CA: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- AWS Secret Access Key: G,W - Account Wrapping key: G,R,W,E - Account key: G,R,W,E - Database Wrapping key: G,R,W,E - User Key: W - User Salt: W - Account Auditor - User password : W - User 2FA device public key: W - Cluster Master Key: R,E - System key: G,R,W,E - SP 800-108 KDF internal state: R,W,E,Z - User Key: W - User Salt: W - Group Administrator - User password : W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- Public key of an outside entity/server (CA): W - API key: G,R - User 2FA device public key: W - JWT Public key of external Application / CA: W - AWS Secret Access Key: G,W - Cluster Master Key: R,E - System key: G,R,W,E - Account Wrapping key: G,R,W,E - Account key: G,R,W,E - Database Wrapping key: G,R,W,E - SP 800-108 KDF internal state: R,W,E,Z - User Key: W - User Salt: W - Group Auditor - User

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						password : W - User 2FA device public key: W - Cluster Master Key: R,E - System key: G,R,W,E - SP 800-108 KDF internal state: R,W,E,Z - User Key: W - User Salt: W Console User - User password : W - Cluster RSA private key for TLS: G,W - Personalization Key: G,E - System key: G,R,W,E - Persisted Sealing Key: G,E - User Key: W - User Salt: W - Cluster RSA public key for TLS: G,W
Random Number	Fetch output	IND_2	None	Random bits	AES-UnAuth	System Administrator

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Generation (Perform approved security functions)	from module DRBG				DRBG ENT-Cond ENT-ESV	r - CTR_DRBG entropy input: G,R,W,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E System Operator - CTR_DRBG entropy input: G,R,W,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E Account Administrator r - CTR_DRBG entropy input: G,R,W,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						V: G,R,W,E - CTR_DRBG Key: G,R,W,E Account Member - CTR_DRBG entropy input: G,R,W,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E Account Auditor - CTR_DRBG entropy input: G,R,W,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E Group Administrator - CTR_DRBG entropy input: G,R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E Group Auditor - CTR_DRBG entropy input: G,R,W,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E Application - CTR_DRBG entropy input: G,R,W,E - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E
Create/Generate key (Perform)	Generate symmetric and/or	IND_2	API request	Key ID	AES-UnAuth DRBG	Account Administrator

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
approved security functions)	asymmetric keys				ECDSA-AsymKeyPair-KeyGen EDDSA-AsymKeyPair-KeyGen LMS-AsymKeyPair-KeyGen RSA-AsymKeyPair-KeyGen SHA-SIMD ENT-Cond ENT-ESV	- CTR_DRBG entropy input: G,R,W,E - HMAC key: G - RSA private key for Digital Signatures: G - RSA private key for Key Encapsulation Operations: G - RSA public key for Key Encapsulation Operations: G - ECDSA private key: G - LMS seed: G,Z - LMS private key: G - Database Wrapping key: E - Symmetric key: G - RSA public key for Digital Signatures: G - CTR_DRBG Seed: G,R,W,E - CTR_DRBG

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						V: G,R,W,E - CTR_DRBG Key: G,R,W,E - ECDSA Public key: G - LMS public key: G - EDDSA private key: G - EDDSA public key: G Group Administrator - CTR_DRBG entropy input: G,R,W,E - HMAC key: G - RSA private key for Digital Signatures: G - RSA private key for Key Encapsulation Operations: G - RSA public key for Key Encapsulation Operations: G - ECDSA private key: G - LMS seed: G,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- LMS private key: G - Database Wrapping key: E - Symmetric key: G - RSA public key for Digital Signatures: G - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E - ECDSA Public key: G - EDDSA private key: G - EDDSA public key: G - LMS public key: G - Application - CTR_DRBG entropy input: G,R,W,E - HMAC key: G - RSA private key for Digital Signatures: G - RSA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						private key for Key Encapsulation Operations: G - RSA public key for Key Encapsulation Operations: G - ECDSA private key: G - LMS seed: G,Z - LMS private key: G - Database Wrapping key: E - Symmetric key: G - RSA public key for Digital Signatures: G - CTR_DRBG Seed: G,R,W,E - CTR_DRBG V: G,R,W,E - CTR_DRBG Key: G,R,W,E - ECDSA Public key: G - EDDSA private key: G - EDDSA public key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G - LMS public key: G
Encrypt/Decrypt (Perform approved security functions)	Symmetric and/or asymmetric encryption and/or decryption of data	IND_2	Key ID, IV, Plaintext or Ciphertext	Ciphertext or Plaintext, IV	AES-Auth AES-UnAuth AES-KTS-Wrap-API	Application - RSA private key for Key Encapsulation Operations: E - Cipher State Wrapping key: G,E - SP 800-108 KDF internal state: R,W - Account key: E - Database Wrapping key: E - Symmetric key: E - RSA public key for Key Encapsulation Operations: E
Sign/Verify (Perform approved security functions)	Create and/or verify digital signatures	IND_2	Key ID, Hash, Data	Signature	DRBG ECDSA-DigSig-SigGen ECDSA-DigSig-SigVer EDDSA-DigSig-SigGen EDDSA-DigSig-SigVer LMS-DigSig-SigGen LMS-	Application - RSA private key for Digital Signatures: E - ECDSA private key: E - ECDSA Public key: E - ECDSA random number "k": G,E,Z - Database

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					DigSig-SigVer RSA-DigSig-SigGen RSA-DigSig-SigVer RSA-DigSig-SigVer (Legacy)	Wrapping key: E - RSA public key for Digital Signatures: E - EDDSA public key: E - EDDSA private key: E - LMS private key: E - LMS public key: E
Wrap/Unwrap (Perform approved security functions)	Wrap and/or unwrap SSPs	IND_2	Key IDs, Wrapped Key, IV	Key ID, Wrapped Key, IV	AES-Auth AES-KTS-Wrap-API RSA-KTS-Wrap	Account Administrator - Database Wrapping key: E - Symmetric key: R,W,E - HMAC key: R,W - RSA private key for Digital Signatures: R,W - RSA private key for Key Encapsulation Operations: R,W,E - ECDSA private key: R,W - EDDSA private key: R,W - LMS private key: W - RSA public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key for Digital Signatures: R,W - RSA public key for Key Encapsulation Operations: R,W,E - EDDSA public key: R,W - ECDSA Public key: R,W - LMS public key: W Group Administrator - Database Wrapping key: E - Symmetric key: R,W,E - HMAC key: R,W - RSA private key for Digital Signatures: R,W - RSA private key for Key Encapsulation Operations: R,W,E - ECDSA private key: R,W - EDDSA private key: R,W - LMS private key: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- RSA public key for Digital Signatures: R,W - RSA public key for Key Encapsulation Operations: R,W,E - ECDSA Public key: R,W - EDDSA public key: R,W - LMS public key: W - Application Database Wrapping key: E - Symmetric key: R,W,E - HMAC key: R,W - RSA private key for Digital Signatures: R,W - RSA private key for Key Encapsulation Operations: R,W,E - ECDSA private key: R,W - EDDSA private key: R,W - LMS private key: W - RSA public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key for Digital Signatures: R,W - RSA public key for Key Encapsulation Operations: R,W,E - ECDSA Public key: R,W - EDDSA public key: R,W - LMS public key: W
MAC (Perform approved security functions)	Calculate CMAC or HMAC digest of data	IND_2	Key ID, Data	MAC	CMAC HMAC SHA	Application - Database Wrapping key: E - HMAC key: E - Symmetric key: E
Digest (Perform approved security functions)	Calculate hash digest of data	IND_2	Data	Digest	SHA	Application
Import Key (Perform approved security functions)	Import a key via TLS	IND_2	Key	Key ID	AES-Auth AES-KTS- Wrap-TLS	Account Administrator - Database Wrapping key: E - HMAC key: W - RSA private key for Digital Signatures: W - RSA private key for Key Encapsulation

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						on Operations: W - ECDSA private key: W - EDDSA private key: W - TLS encryption key: G,E,Z - Symmetric key: W - LMS private key: W - RSA public key for Digital Signatures: W - RSA public key for Key Encapsulati on Operations: W - ECDSA Public key: W - EDDSA public key: W - LMS public key: W Group Administrato r - Database Wrapping key: E - HMAC key: W - RSA private key for Digital Signatures: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- RSA private key for Key Encapsulation Operations: W - ECDSA private key: W - EDDSA private key: W - TLS encryption key: G,E,Z - Symmetric key: W - LMS private key: W - RSA public key for Digital Signatures: W - RSA public key for Key Encapsulation Operations: W - ECDSA Public key: W - EDDSA public key: W - LMS public key: W - Application Database Wrapping key: E - HMAC key: W - RSA private key for Digital

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Signatures: W - RSA private key for Key Encapsulation Operations: W - ECDSA private key: W - EDDSA private key: W - TLS encryption key: G,E,Z - Symmetric key: W - LMS private key: W - RSA public key for Digital Signatures: W - RSA public key for Key Encapsulation Operations: W - ECDSA Public key: W - EDDSA public key: W - LMS public key: W
Export Key (Perform approved security functions)	Export Key via TLS	IND_2	Key ID	Key	AES-Auth AES-KTS- Wrap-TLS	Account Administrator - Database Wrapping key: E - HMAC key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						R - RSA private key for Digital Signatures: R - RSA private key for Key Encapsulation Operations: R - ECDSA private key: R - EDDSA private key: R - TLS encryption key: G,E,Z - Symmetric key: R - RSA public key for Digital Signatures: R - RSA public key for Key Encapsulation Operations: R - ECDSA Public key: R - EDDSA public key: R Group Administrator - Database Wrapping key: E - HMAC key: R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- RSA private key for Digital Signatures: R - RSA private key for Key Encapsulation Operations: R - ECDSA private key: R - EDDSA private key: R - TLS encryption key: G,E,Z - Symmetric key: R - RSA public key for Digital Signatures: R - RSA public key for Key Encapsulation Operations: R - ECDSA Public key: R - EDDSA public key: R - Application - Database Wrapping key: E - HMAC key: R - RSA private key for Digital

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Signatures: R - RSA private key for Key Encapsulation Operations: R - ECDSA private key: R - EDDSA private key: R - TLS encryption key: G,E,Z - Symmetric key: R - RSA public key for Digital Signatures: R - RSA public key for Key Encapsulation Operations: R - ECDSA Public key: R - EDDSA public key: R
Run self-tests (Perform self-tests)	Execute the module's pre-operational and conditional self-tests	IND_1	CLI commands	CLI Status	AES-Auth AES-UnAuth DRBG ECDSA-DigSig- SigGen ECDSA-DigSig- SigVer EDDSA-DigSig-	Console User

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					SigGen EDDSA- DigSig- SigVer HMAC KAS-ECC- SSC KAS-FFC- SSC KBKDF RSA-KTS- Wrap LMS- AsymKeyP air-KeyGen LMS- DigSig- SigGen LMS- DigSig- SigVer PBKDF RSA- DigSig- SigGen RSA- DigSig- SigVer SHA SHA-SIMD XOF TLS-KAS- 135KDF ENT-Cond ENT-ESV RSA- DigSig- SigVer (Legacy)	
Get status (Show Status)	Query the status and version information of the module	IND_1, IND_2	CLI commands, API and GUI requests	CLI, API, and GUI responses	None	System Administrator System Operator Account Administrator Account Member

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Account Auditor Group Administrator Group Auditor Application Console User
Authentication (Perform approved security functions)	Authenticate with the module	IND_1, IND_2	Authentication material	CLI, API, and GUI responses	AES-Auth ECDSA-DigSig-SigVer HMAC PBKDF RSA-DigSig-SigVer SHA RSA-DigSig-SigVer (Legacy)	System Administrator - User password : W - Bearer token: W - User 2FA device public key: E - User Key: W - User Salt: W System Operator - User password : W - Bearer token: W - User 2FA device public key: E - User Key: W - User Salt: W Account Administrator - User password : W - Bearer token: W - User 2FA device

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						public key: E - User Key: W - User Salt: W Account Member - User password : W - Bearer token: W - User 2FA device public key: E - User Key: W - User Salt: W Account Auditor - User password : W - Bearer token: W - User 2FA device public key: E - User Key: W - User Salt: W Group Administrator - User password : W - Bearer token: W - User 2FA device public key: E - User Key: W - User Salt: W Group

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Auditor - User password : W - Bearer token: W - User 2FA device public key: E - User Key: W - User Salt: W Application - API key: W - AWS Secret Access Key: W - Public key of an outside entity/server (CA): E - RSA Public key of external Application: E - JWT Public key of external Application / CA: E - Bearer token: W Console User - Console User Password : W - Console User Salt: W - Console User Key: W

Table 13: Approved Services

The module provides the services detailed in the preceding table. For the indicators, the module provides information in the following forms:

- IND_1: The module's CLI provides indicators in the form of human-readable outputs, with success indicators being output to the 'stdout' output stream and error indicators being output to the 'stderr' output stream.
- IND_2: The module's REST API and Web GUI provide indicators in the form of HTTP status codes. An HTTP status code of 2xx is indicative of success, while 4xx and 5xx status codes indicate errors. The HTTP status codes are also delivered alongside human-readable outputs. Full details of this indicator can be found in the Fortanix DSM REST API specification [API¹].

4.4 Non-Approved Services

The module does not have any non-approved services.

4.5 External Software/Firmware Loaded

The module can load firmware from external sources in the following manners. Both loading mechanisms are **not** complete image replacements based on FIPS 140-3 IG 10.3.F, as they do not replace the existing module components in their entirety.

- The module's firmware can be updated via the Web GUI, where the new firmware images are signed using ECDSA (P-256 with SHA-256 as the hashing function) and verified by the module using an authentication key that's loaded independently of the firmware load process (namely, it is included in the previous version's image).
- An operator can load plugins via the Web GUI or REST API, where the plugins are individually signed using ECDSA (P-256, P-384, or P-521) or RSA (2048, 3072, 4096) with SHA2-256, SHA2-384, or SHA-512 and verified using an authentication key that's loaded independently of the plugin loading process (namely, it is loaded as part of setting up the plugin code signing policy).

If an operator opts to load new firmware or a plugin onto the module, the module would no longer be FIPS 140-3 validated unless the new firmware or plugin has been separately FIPS 140-3 validated. Only firmware and plugin versions listed in this Security Policy are FIPS 140-3 validated.

5 Software/Firmware Security

5.1 Integrity Techniques

The module's integrity is checked at boot time using:

- HMAC-SHA2-256 with a 256-bit key is used to check the integrity of the backend components.
- A 16-bit sum-of-bytes checksum is used to check the integrity of the personalization key store's executable code.
- A 32-bit CRC32C checksum is used to check the integrity of filesystems on the module.

¹ API's can be found here: <https://support.fortanix.com/apidocs>

The module is delivered as a self-extracting tar archive which contains all firmware images and packages associated with a particular release. The module is comprised of both compiled code as well as uncompiled code (namely scripts where the interpreter, configuration and environmental dependencies are included as well). This scenario of uncompiled code is acceptable per CMVP guidance. ([Guidance about Code in "executable form" - Projects](#))

5.2 Initiate on Demand

An operator can initiate the integrity check on demand by rebooting the module via the CLI.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

How Requirements are Satisfied:

This section is designated as N/A, as the operational environment is a limited operational environment since the module is designed to accept only controlled firmware changes that successfully pass the firmware load tests (see section 4.5 External Software/Firmware Loaded).

Per the CMVP Management Manual: *Section 6.6, Operational Environment may be designated as N/A if the operational environment for the cryptographic module is a limited or non-modifiable operational environment and Section 6.7, Physical Security is greater than Security Level 1 (AS06.04).*

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Tamper-Evident Seals	At least monthly	The DSM appliance enclosure should be periodically checked by the Crypto Officer for evidence of tampering damage to the two (FX2200's) or six (FX3400's and FX3500's) tamper-evident labels and any physical damage to the enclosure material. The frequency of a physical inspection depends upon the information being protected and the environment in which the unit is located. At a minimum, it would be expected that a physical inspection would be made by the Crypto Officer at least monthly. The tamper evident labels are applied at the Fortanix manufacturing facility, are serialized, and are not available for order or replacement from Fortanix. The labels are designed and intended to stay intact for the entire life of the module.
Strong Enclosure	At least monthly	The DSM appliance enclosure should be periodically checked by the Crypto Officer for evidence of tampering

Mechanism	Inspection Frequency	Inspection Guidance
		damage to the three tamper-evident labels and any physical damage to the enclosure material. The frequency of a physical inspection depends upon the information being protected and the environment in which the unit is located. At a minimum, it would be expected that a physical inspection would be made by the Crypto Officer at least monthly.
Tamper Response and Zeroization Circuitry	Never	This is internal to the module's enclosure and can't be inspected without triggering zeroization.

Table 14: Mechanisms and Actions Required

The table above summarizes the inspection / testing of physical security mechanisms. The following guidelines should be considered when producing an Operational Policy for the environment for which the module is deployed.

The DSM appliance enclosure should be periodically checked by the Crypto Officer for evidence of tampering damage to the four tamper-evident labels and any physical damage to the enclosure material.

The frequency of a physical inspection depends upon the information being protected and the environment in which the unit is located. At a minimum, it would be expected that a physical inspection would be done by the Crypto Officer at least monthly.

The tamper evident labels are applied at the Fortanix manufacturing facility, are serialized, and are not available for order or replacement from Fortanix. The labels are designed and intended to stay intact for the entire life of the module. The labels are applied in the three positions shown in the figure below. If labels are found to be tampered, the module is to be returned to Fortnaix.

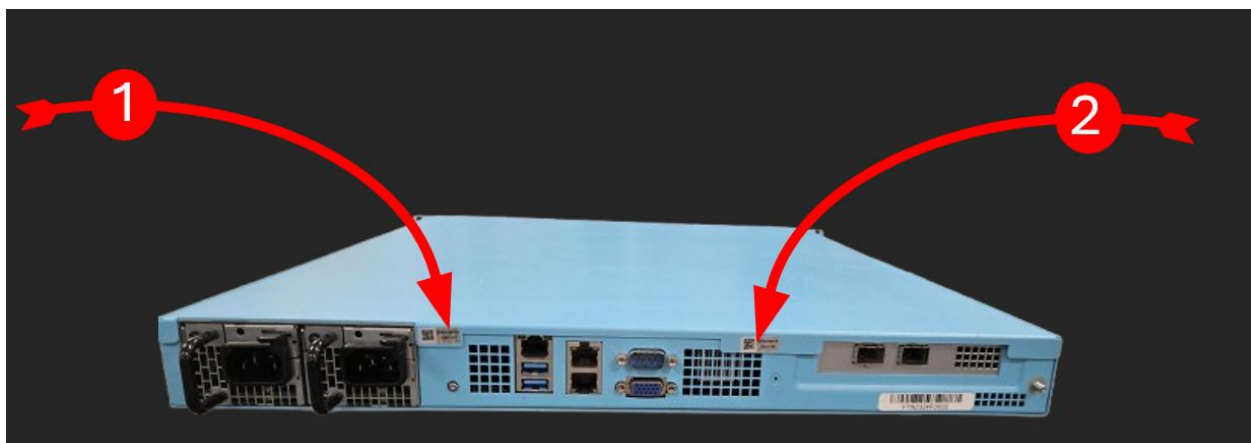


Figure 12: Tamper Evident Label Positions on FX2200-II-TN-F, FX2200-II-SXN-F, FX3400-T, FX3400-F, FX3500-T and FX3500-F



Figure 13: Tamper Evident Label Positions on FX3400-T, FX3400-F, FX3500-T and FX3500-F

The following figure shows the tamper label. It leaves “VOID” markings in place of tamper label and the tamper label cannot be reapplied.

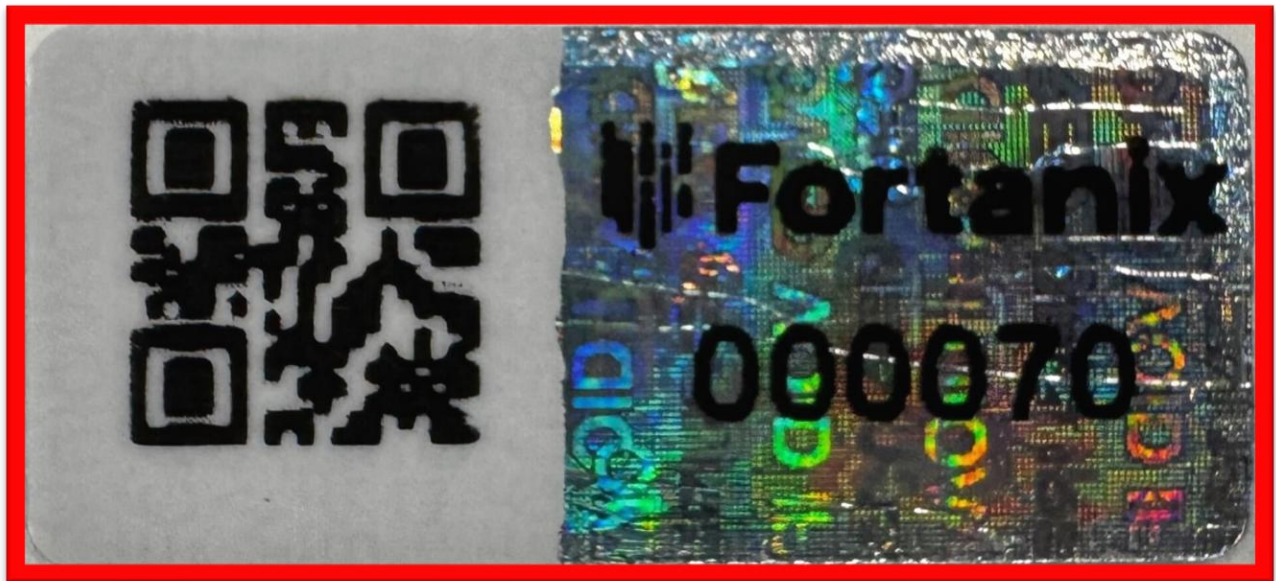


Figure 14: Tamper Evident Label

The two tamper seals sit over a screw on the lid and extend over the lid seam to the module chassis, as shown in the figure below. The only way to remove the cover is to break or damage the tamper seals.

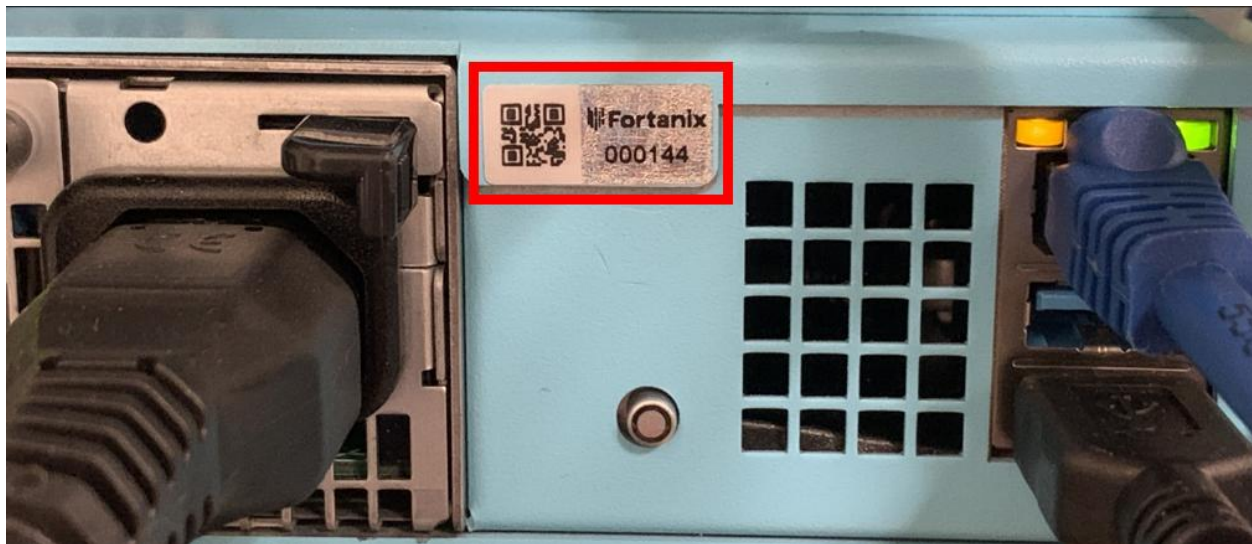


Figure 15: Tamper Evident Label Closeup

7.5 EFP/EFT Information

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature	Below 0 Celsius	EFP	Shutdown
HighTemperature	Above 50 Celsius	EFP	Shutdown
LowVoltage	Below 70 Volts (AC)	EFT	Shutdown
HighVoltage	Above 389-410 Volts (AC)	EFT	Shutdown

Table 15: EFP/EFT Information

The module implements EFP/EFT features that will force the module to shut down at the extremes detailed in the preceding table.

The module is intended to be deployed in an environment with the following parameters, and the enclosure will maintain its strength and hardness characteristics over the following ranges:

- **Voltage / Frequency:** AC input: 100-240v 63-47hz 5-2.5a
- **Temperature:** Operating: 0 C° to 50 C° / Storage and Distribution -40 C° to 70 C°

7.6 Hardness Testing Temperature Ranges

Temperature Type	Temperature
LowTemperature	-40 Celsius
HighTemperature	70 Celsius

Table 16: Hardness Testing Temperatures

The module's hardness is evaluated at the lowest and highest temperatures within the module's intended temperature range of storage (see preceding table).

7.7 Additional Information

The cryptographic module is a multi-chip standalone product that consists of production-grade components. The strong enclosure of the cryptographic module is opaque within the visible spectrum and is designed to prevent the gathering of information of the module's internal construction or components. The removable covers are protected with tamper-evident seals. The tamper-evident seals must be checked periodically by the Crypto Officer. If the tamper-evident seals are broken or missing, the Crypto Officer must halt the operation of the module and ship the module to Fortanix for replacement.

The module contains tamper response and zeroization circuitry. The tamper response and zeroization circuitry immediately (in less than 1 second) zeroizes all plaintext secret and private keys and CSPs when a cover is removed. The tamper response and zeroization circuitry remains operational when plaintext secret and private cryptographic keys or CSPs are contained within the cryptographic module.

Ventilation holes are constructed in a manner that prevents undetected physical probing inside the enclosure.

8 Non-Invasive Security

Non-invasive security is N/A as there are currently no requirements in SP 800-140F.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
USB Key Store	The module has an internal, battery-backed USB key store used for SSP storage (plaintext format)	Static
SSDs	The module has multiple SSDs used for SSP and other data storage, both in encrypted and plaintext formats	Static
RAM	The module has RAM used for SSP and other data storage, both in encrypted and plaintext formats	Dynamic

Table 17: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
TLS Input	Operator	Module	Encrypted	Automated	Electronic	
TLS Output	Module	Operator	Encrypted	Automated	Electronic	
AES Wrapped Input	Operator	Module	Encrypted	Automated	Electronic	AES-Auth

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
AES Wrapped Output	Module	Operator	Encrypted	Automated	Electronic	AES-KTS-Wrap-API
RSA Wrapped Input	Operator	Module	Encrypted	Automated	Electronic	RSA-KTS-Wrap
RSA Wrapped Output	Module	Operator	Encrypted	Automated	Electronic	RSA-KTS-Wrap
Plaintext Input	Operator	Module	Plaintext	Automated	Electronic	
Plaintext Output	Module	Operator	Plaintext	Automated	Electronic	
CLI Input	Operator	Module	Plaintext	Manual	Direct	
Factory	Manufacturer	Module	Plaintext	N/A	N/A	

Table 18: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Persistent Overwrite	Keys stored on the module's USB Key Store (the Personalization Key) are overwritten with zeroes upon execution of the zeroize command. Success is indicated via the message "Zeroization successfully succeeded!"	Overwriting with zeros is an acceptable form of zeroization per ISO/IEC 19790:2012 section 7.9.7. This is performed in a time (<1 second) that is not sufficient to compromise plaintext SSPs.	CLI command for zeroization
Cryptographic	SSPs stored persistently on the module's SSD are encrypted using AES-GCM that can be zeroized via overwrite. This renders the SSPs that were encrypted irretrievable	Encrypted SSPs are considered "protected" per ISO/IEC 19790:2012 section 3.36, and thus direct zeroization is not required (section 7.9.7) This is performed in a time (<1 second) that is not sufficient to compromise plaintext SSPs.	CLI command for zeroization
Ephemeral Overwrite	SSPs stored in the module's RAM are directly overwritten with zeroes when they are no longer needed	Overwriting with zeros is an acceptable form of zeroization per ISO/IEC 19790:2012 section 7.9.7. This is performed in a time (<100 ms) that is not	Automatically performed when SSP is no longer needed

Zeroization Method	Description	Rationale	Operator Initiation
		sufficient to compromise SSPs.	
Ephemeral Reboot	SSPs stored in the module's RAM	Overwriting with zeros is an acceptable form of zeroization per ISO/IEC 19790:2012 section 7.9.7. The module's input/output interfaces become unresponsive in <1s during these processes, which as a whole can take around 5 minutes to complete. This is not a sufficient amount of time to compromise SSPs due to the physical security mechanisms in place.	CLI command for zeroization Powering off, Power-cycling or rebooting the module wipes the contents of the module's RAM

Table 19: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Firmware update key	Public key used to validate the signature of firmware update	P-256 (256 bits) - 128 bits	ECDSA - PSP			ECDSA-DigSig-SigVer
Plugin signing key	Key used to validate the signature of loaded plugins. RSA (2048, 3072, 4096) ECDSA (P-256, P-384, P-521)	256 - 4096 bits - 112 - 256 bits	RSA, ECDSA - PSP			ECDSA-AsymKey Pair-PubKeyVal ECDSA-DigSig-SigVer RSA-DigSig-SigVer RSA-DigSig-SigVer (Legacy)
Personalization Key	This key is used to derive the persisted sealing key	256 bits - 256 bits	AES - CSP	DRBG		AES-Auth KBKDF

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Persisted Sealing Key	AES 256 bits This key is used to wrap top level keys (Cluster Master key and node private key)	256 bits - 256 bits	AES - CSP		KBKDF	AES-Auth
Cluster Master Key	Key Derivation Key 256 bits This key is used to derive System key and Account Wrapping key	256 bits - 256 bits	SP 800-108 KDF (Key derivation key) - CSP	DRBG		KBKDF
System key	AES 256 bits This key is used to wrap all user and session information that is stored in persistent storage	256 bits - 256 bits	AES - CSP		KBKDF	AES-Auth
Account Wrapping key	AES 256 bits This key is used to wrap Account key when it is stored in persistent storage	256 bits - 256 bits	AES - CSP		KBKDF	AES-Auth
Account key	Key Derivation Key 256-bits This key is used to derive Database Wrapping key and Cipher State Wrapping key	256 bits - 256 bits	SP 800-108 KDF (Key derivation key) - CSP	DRBG		KBKDF
Database Wrapping key	AES 256 bits This key is used to wrap all account / tenant data and keys that belong to a specific account / tenant when it is stored in persistent storage	256 bits - 256 bits	AES GCM - CSP		KBKDF	AES-Auth
Cipher State Wrapping key	AES 128 bits This key is used to wrap all cipher state data that belong to a specific account / tenant	128 bits - 128 bits	AES-GCM - CSP		KBKDF	AES-Auth

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Symmetric key	AES - 128, 192, 256	128 - 256	AES - CSP	DRBG		AES-Auth AES-UnAuth AES-KTS-Wrap-API CMAC
HMAC key	HMAC-SHA-1: 112-bit minimum key HMAC-SHA2-256: 128-bit minimum key HMAC-SHA2-384: 192-bit minimum key HMAC-SHA2-512: 256-bit minimum key	>= 112 bits - >= 112 bits	HMAC - CSP	DRBG		HMAC
RSA private key for Digital Signatures	RSA - 2048 to 8192 This key is used for Digital Signature Generation and Verification	2048 - 8192 bits - 112 - 202 bits	RSA - CSP	DRBG RSA-AsymKey Pair- KeyGen		RSA-DigSig-SigGen
RSA public key for Digital Signatures	RSA-2048 to 8192 used for Digital Signature Generation and Verification	2048-8192 - 112-202 bits	RSA - PSP	DRBG RSA-AsymKey Pair- KeyGen		RSA-DigSig-SigVer RSA-DigSig-SigVer (Legacy)
RSA private key for Key Encapsulation Operations	RSA - 2048 to 8192 This key is used for Key Unencapsulation operations	2048 - 8192 bits - 112 - 202 bits	RSA - CSP	DRBG RSA-AsymKey Pair- KeyGen		RSA-KTS-Wrap
RSA public key for Key Encapsulation Operations	RSA-2048 to 8192 This key is used for Key encapsulation operations	2048-8192 bits - 112-202 bits	RSA - PSP	DRBG RSA-AsymKey Pair- KeyGen		RSA-KTS-Wrap
Cluster RSA private key for TLS	RSA - 2048 bits	2048 bits - 112 bits	RSA - CSP	DRBG RSA-AsymKey Pair- KeyGen		RSA-DigSig-SigGen

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Cluster RSA public key for TLS	RSA- 2048 bits	2048 bits - 112 bits	RSA - PSP	DRBG RSA-AsymKey Pair-KeyGen		RSA-DigSig-SigVer RSA-DigSig-SigVer (Legacy)
ECDSA private key	EC - P-224, P-256, P-384, P-521	224 - 521 bits - 112 - 256 bits	ECDSA - CSP	DRBG ECDSA-AsymKey Pair-KeyGen		ECDSA-AsymKey Pair-PubKeyVal ECDSA-DigSig-SigGen
ECDSA Public key	EC-P-224, P-256, P-384, P-521	224- 521 bits - 112- 256 bits	ECDSA - PSP	DRBG ECDSA-AsymKey Pair-KeyGen		ECDSA-AsymKey Pair-PubKeyVal ECDSA-DigSig-SigVer
EDDSA private key	ED-25519	255 bits - 128 bits	EDDSA - CSP	DRBG EDDSA-AsymKey Pair-KeyGen		EDDSA-DigSig-SigGen
EDDSA public key	ED-25519	255 bits - 128 bits	EDDSA - PSP	DRBG EDDSA-AsymKey Pair-KeyGen		EDDSA-DigSig-SigVer
ECDSA random number "k"	k = 224-bits (P-224) k = 256-bits (P-256) k = 384-bits (P-384) k = 521-bits (P-521)	224 - 521 bits - 112 - 256 bits	ECDSA - CSP	DRBG ECDSA-AsymKey Pair-KeyGen		ECDSA-DigSig-SigGen
LMS private key	LMS_SHA256_M24_H5 LMS_SHA256_M24_H10 LMS_SHA256_M24_H15 LMS_SHA256_M24_H20	20KB - 1.1GB - 192, 256 bits	LMS - CSP	DRBG LMS-AsymKey Pair-KeyGen		LMS-DigSig-SigGen

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	LMS_SHA256_M32_H5 LMS_SHA256_M32_H10 LMS_SHA256_M32_H15 LMS_SHA256_M32_H20					
LMS public key	LMS_SHA256_M24_H5 LMS_SHA256_M24_H10 LMS_SHA256_M24_H15 LMS_SHA256_M24_H20 LMS_SHA256_M32_H5 LMS_SHA256_M32_H10 LMS_SHA256_M32_H15 LMS_SHA256_M32_H20	20 KB-1.1 GB - 192-256 bits	LMS - PSP	DRBG LMS-AsymKey Pair-KeyGen		LMS-DigSig-SigVer
LMS seed	24 - 32-byte seed used to generate LMS key pairs	192, 256 bits - 192, 256 bits	LMS - CSP	DRBG LMS-AsymKey Pair-KeyGen		LMS-AsymKey Pair-KeyGen
ECCDH private key	EC - P-224, P-256, P-384, P-521	224 - 521 bits - 112 - 256 bits	ECDH - CSP	DRBG ECDSA-AsymKey Pair-KeyGen		ECDSA-AsymKey Pair-PubKeyVal KAS-ECC-SSC TLS-KAS-Full
ECCDH public key	EC - P-224, P-256, P-384, P-521	224-521 bits - 112-256 bits	ECDH - PSP	DRBG ECDSA-AsymKey Pair-KeyGen		ECDSA-AsymKey Pair-PubKeyVal KAS-ECC-SSC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						TLS-KAS-Full
FFCDH private key	FFC - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	2048 - 8192 bits - 112 - 202 bits	DH - CSP	DRBG SafePrime s-AsymKey Pair-KeyGen		KAS-FFC-SSC SafePrime s-AsymKey Pair-KeyVer TLS-KAS-Full
FFCDH public key	FFC - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	2048-8192 bits - 112-202 bits	DH - PSP	DRBG SafePrime s-AsymKey Pair-KeyGen		KAS-FFC-SSC SafePrime s-AsymKey Pair-KeyVer TLS-KAS-Full
TLS KDF internal state	TLS v1.2 or v1.3 KDF (HMAC-SHA2-256 PRF or HMAC-SHA2-384 PRF)	variable size, >= 256 bits - 256 bits	State - CSP		KAS-ECC-SSC KAS-FFC-SSC	TLS-KAS-135KDF TLS-KAS-Full
TLS integrity key	HMAC-SHA2-256 (256-bit key) HMAC-SHA2-384 (384-bit key)	256 - 384 bits - 256 - bits	HMAC - CSP		TLS-KAS-135KDF	HMAC
TLS encryption key	AES-128-GCM, AES-256-GCM	128, 256 bits - 128, 256 bits	AES - CSP		TLS-KAS-135KDF	AES-Auth AES-KTS-Wrap-TLS
TLS pre-master secret	Shared Secret	variable size, >= 224 bits - >= 112 bits	Shared Secret - CSP		KAS-ECC-SSC KAS-FFC-SSC	TLS-KAS-135KDF TLS-KAS-Full
TLS extended master secret	Shared Secret	384 bits - 256 bits	Shared Secret - CSP		TLS-KAS-135KDF	TLS-KAS-135KDF TLS-KAS-Full

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SP 800-108 KDF internal state	SP 800-108 KDF in Feedback Mode (§5.2) with HMAC-SHA2-256	256 bits - 256 bits	State - CSP		KBKDF	KBKDF
User password	String of ASCII characters	Minimum 8 bytes - >= 64 bits	Password - CSP			PBKDF
Console User Password	String of ASCII characters	Minimum 8 bytes - >= 64 bits	Password - CSP			PBKDF
API key	Application Authentication Data	64 bytes - 256 bits	Memorized Secret - CSP	DRBG		
AWS Secret Access Key	HMAC-SHA2-256 key used as Authentication Data	32 bytes - 256 bits	HMAC - CSP	DRBG		HMAC
Bearer token	Authentication Data	64 bytes - 256 bits	Authentication Data - CSP	DRBG		
Public key of an outside entity/server (CA)	RSA - 2048 to 8192 RSA Key used for authentication using digital certificate	2048 - 8192 bits - 112 - 202 bits	RSA - PSP			RSA-DigSig-SigVer RSA-DigSig-SigVer (Legacy)
RSA Public key of external Application	RSA - 2048 to 8192 RSA Key used for authentication using digital certificate	2048 - 8192 bits - 112 - 202 bits	RSA - PSP			RSA-DigSig-SigVer RSA-DigSig-SigVer (Legacy)
User 2FA device public key	EC P-256 Authentication Data	256 bits - 128 bits	ECDSA - PSP			ECDSA-AsymKeyPair-PubKeyVal ECDSA-

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						DigSig-SigVer
JWT Public key of external Application / CA	RSA - 2048 to 4096 ECDSA - P-256, P-384, P-521 Authentication Data	256 - 4096 bits - 112 - 256 bits	RSA, ECDSA - PSP			ECDSA-AsymKey Pair-PubKeyVal ECDSA-DigSig-SigVer RSA-DigSig-SigVer RSA-DigSig-SigVer (Legacy)
User Key	Output of PBKDF used for authentication	256 bits - 256 bits	Authentication Data - CSP		PBKDF	
Console User Key	Output of PBKDF used for authentication	256 bits - 256 bits	Authentication Data - CSP		PBKDF	
User Salt	Salt used in PBKDF to derive User Key	256 bits - 256 bits	Authentication Data - CSP	DRBG		PBKDF
Console User Salt	Salt used in PBKDF to derive Console User Key	256 bits - 256 bits	Authentication Data - CSP	DRBG		PBKDF
CTR_DRBG entropy input	Entropy input per IG D.L	384 bits - 384 bits	DRBG - CSP	DRBG ENT-ESV		DRBG
CTR_DRBG Seed	DRBG Seed per IG D.L	384 bits - 384 bits	DRBG - CSP	DRBG ENT-ESV		DRBG
CTR_DRBG V	DRBG V per IG D.L	128 bits - 128 bits	DRBG - CSP	DRBG ENT-ESV		DRBG
CTR_DRBG Key	DRBG Key per IG D.L	256 bits -	DRBG - CSP	DRBG ENT-ESV		DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		256 bits				

Table 20: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Firmware update key	Factory	SSDs:Plaintext			
Plugin signing key	TLS Input TLS Output	SSDs:Encrypted		Cryptographic	
Personalization Key		USB Key Store:Plaintext		Cryptographic	Persisted Sealing Key:Derives
Persisted Sealing Key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	Personalization Key:Derived From RSA private key for Digital Signatures:Wraps
Cluster Master Key		SSDs:Encrypted		Cryptographic	Persisted Sealing Key:Wrapped By Account Wrapping key:Derives System key:Derives
System key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	Plugin signing key:Wraps Cluster Master Key:Derived From Cluster RSA key pair for TLS:Wraps User password :Wraps Bearer token:Wraps User 2FA device public key:Wraps
Account Wrapping key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	Cluster Master Key:Derived From Account key:Wraps
Account key		SSDs:Encrypted		Cryptographic	Account Wrapping key:Wrapped by

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					Database Wrapping key:Derives Cipher State Wrapping key:Derives
Database Wrapping key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	Account key:Derived From Symmetric key:Wraps RSA private key for Digital Signatures:Wraps RSA key pair for Key Encapsulation Operations:Wraps EDDSA key pair:Wraps LMS key pair:Wraps API key:Wraps
Cipher State Wrapping key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	Account key:Derived From
Symmetric key	TLS Input TLS Output AES Wrapped Input AES Wrapped Output RSA Wrapped Input RSA Wrapped Output	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By
HMAC key	TLS Input TLS Output AES	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	Wrapped Input AES Wrapped Output RSA Wrapped Input RSA Wrapped Output				
RSA private key for Digital Signatures	TLS Input TLS Output AES Wrapped Input AES Wrapped Output RSA Wrapped Input RSA Wrapped Output	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By
RSA public key for Digital Signatures	TLS Input TLS Output AES Wrapped Input AES Wrapped Output RSA Wrapped Input RSA Wrapped Output	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By
RSA private key for Key Encapsulation Operations	TLS Input TLS Output	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	AES Wrapped Input AES Wrapped Output RSA Wrapped Input RSA Wrapped Output				
RSA public key for Key Encapsulation Operations	TLS Input TLS Output AES Wrapped Input AES Wrapped Output RSA Wrapped Input RSA Wrapped Output	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By
Cluster RSA private key for TLS		SSDs:Encrypted		Cryptographic	System key:Wrapped by
Cluster RSA public key for TLS	Plaintext Output	SSDs:Plaintext		Cryptographic	System key:Wrapped By
ECDSA private key	TLS Input TLS Output AES Wrapped Input AES Wrapped Output RSA Wrapped Input	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	RSA Wrapped Output				
ECDSA Public key	TLS Input TLS Output AES Wrapped Input AES Wrapped Output RSA Wrapped Input RSA Wrapped Output	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By
EDDSA private key	TLS Input TLS Output AES Wrapped Input AES Wrapped Output RSA Wrapped Input RSA Wrapped Output	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By
EDDSA public key	TLS Input TLS Output AES Wrapped Input AES Wrapped Output RSA Wrapped	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	d Input RSA Wrapped Output				
ECDSA random number "k"		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	
LMS private key	TLS Input AES Wrapped Input RSA Wrapped Input	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped By LMS seed:Generated By
LMS public key	TLS Input TLS Output AES Wrapped Input AES Wrapped Output RSA Wrapped Input RSA Wrapped Output	SSDs:Encrypted		Cryptographic	
LMS seed		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	TLS pre-master secret:Establishes
ECCDH private key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	TLS pre-master secret:Establishes
ECCDH public key	Plaintext Output	RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	TLS pre-master secret:Establishes
FFCDH private key		RAM:Plaintext	After Usage	Ephemeral Overwrite	TLS pre-master secret:Establishes

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Ephemeral Reboot	
FFCDH public key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	TLS pre-master secret:Establishes
TLS KDF internal state		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	
TLS integrity key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	TLS extended master secret:Derived By
TLS encryption key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	TLS extended master secret:Derived By
TLS pre-master secret		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	TLS extended master secret:Derived By ECCDH key pair:Established By FFCDH key pair:Established By
TLS extended master secret		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	TLS pre-master secret:Derived By TLS integrity key:Derives TLS encryption key:Derives
SP 800-108 KDF internal state		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	
User password	TLS Input	RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	
Console User Password	CLI Input	RAM:Plaintext		Ephemeral Overwrite Ephemeral Reboot	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
API key	TLS Input TLS Output	SSDs:Encrypted		Cryptographic	Database Wrapping key:Wrapped by
AWS Secret Access Key	TLS Input TLS Output	SSDs:Encrypted		Cryptographic	System key:Wrapped by
Bearer token	TLS Input TLS Output	SSDs:Encrypted		Cryptographic	System key:Wrapped by
Public key of an outside entity/server (CA)	TLS Input Plaintext Input	RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	
RSA Public key of external Application	TLS Input Plaintext Input	RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	
User 2FA device public key	TLS Input	SSDs:Encrypted		Cryptographic	System key:Wrapped by
JWT Public key of external Application / CA	TLS Input TLS Output	SSDs:Encrypted		Cryptographic	System key:Wrapped by
User Key		SSDs:Encrypted		Cryptographic	User password :Derived By User Salt:Derived By System key:Wrapped By
Console User Key		SSDs:Encrypted		Cryptographic	Console User Password :Derived By Console User Salt:Derived By Personalization Key:Wrapped By
User Salt		SSDs:Encrypted		Cryptographic	User Key:Derives
Console User Salt		SSDs:Encrypted		Cryptographic	Console User Key:Derives
CTR_DRBG entropy input		RAM:Plaintext	After Usage	Ephemeral Overwrite	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Ephemeral Reboot	
CTR_DRBG Seed		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	
CTR_DRBG V		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	
CTR_DRBG Key		RAM:Plaintext	After Usage	Ephemeral Overwrite Ephemeral Reboot	

Table 21: SSP Table 2

The module does not support the import or export of unprotected CSPs. The preceding tables describe cryptographic keys and CSPs used by the module.

9.5 Transitions

SHA-1 is non-approved for digital signature generation and deprecated through December 31, 2030 for non-digital signature generation.

The minimum approved security strength will transition from 112 bits to 128 bits after the year 2030 according to SP 800-57 Part 1 Rev 5 Table 4.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Firmware Integrity Test	256-bit key size	HMAC-SHA2-256	SW/FW Integrity	Success: "Software Integrity test succeeded" Error: "Software integrity check failed"	MAC is calculated and verified over binary executables
Key Store Integrity Test		16-bit Sum-of-bytes Checksum	SW/FW Integrity	Success: "Software integrity check of personalization key store module succeeded" Error:	Checksum is calculated and verified over personalization key store binary

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
				"Integrity check failed"	
Filesystem Integrity Test		32-bit CRC32C Checksum	SW/FW Integrity	Success: "no error found" Error: "checksum error"	Checksum is calculated and verified over specific filesystems

Table 22: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES Encrypt	128-bit key size in ECB, CBC, CFB128, OFB, and CTR Modes 192-bit key size in ECB, CBC, CFB128, and OFB Modes 256-bit key size in ECB, CBC, CFB128, and OFB Modes	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "AES self test failed"	Encrypt	Power On Crypto Library Load
AES Decrypt	128-bit key size in ECB, CBC, CFB128, OFB, and CTR Modes 192-bit key size in ECB, CBC, CFB128, and OFB Modes 256-bit key size in ECB, CBC, CFB128, and OFB Modes	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "AES self test failed"	Decrypt	Power On Crypto Library Load
AES GCM Encrypt	AES GCM 128-bit, 192-bit, and 256-bit key size	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "GCM self test failed"	Encrypt	Power On Crypto Library Load

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES GCM Decrypt	AES GCM 128-bit, 192-bit, and 256-bit key size	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "GCM self test failed"	Decrypt	Power On Crypto Library Load
AES CCM Encrypt	AES CCM 128-bit key size AES CCM CAST (encryption and decryption)	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "CCM self test failed"	Encrypt	Power On Crypto Library Load
AES CCM Decrypt	AES CCM 128-bit key size AES CCM CAST (encryption and decryption)	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "CCM self test failed"	Decrypt	Power On Crypto Library Load
AES Key Wrap	AES-KW and AES-KWP 128-bit, 192-bit, 256-bit key sizes	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "KW/KWP self test failed"	Encrypt	Power On Crypto Library Load
AES Key Unwrap	AES-KW and AES-KWP 128-bit, 192-bit, 256-bit key sizes	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "KW/KWP self test failed"	Decrypt	Power On Crypto Library Load
AES CMAC Encrypt	AES CMAC 128-bit, 192-bit, 256-bit key sizes	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "CMAC self test failed"	Encrypt	Power On Crypto Library Load
AES CMAC Decrypt	AES CMAC 128-bit, 192-bit, 256-bit key sizes	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "CMAC self test failed"	Decrypt	Power On Crypto Library Load
AES FF1 Encrypt	AES FF1 128-bit key size Radix 36	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "FF1 self test failed"	Encrypt	Power On Crypto Library Load
AES FF1 Decrypt	AES FF1 128-bit key size Radix 36	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "FF1 self test failed"	Decrypt	Power On Crypto Library Load

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Counter DRBG (A534 4)	Counter DRBG with AES-256	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "CTR DRBG self test failed"	Generate	Power On Crypto Library Load
ECDSA SigGen (FIPS186-5) (A534 4)	P-224 curve, SHA2-256	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "ECDSA self test failed"	Sign	Power On Crypto Library Load
ECDSA SigVer (FIPS186-5) (A534 4)	P-224 curve, SHA2-256	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "ECDSA self test failed"	Verify	Power On Crypto Library Load
EDDSA SigGen (A534 4)	ED-25519	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "EdDSA self test failed"	Sign	Power On Crypto Library Load
EDDSA SigVer (A534 4)	ED-25519	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "EdDSA self test failed"	Verify	Power On Crypto Library Load
HMAC-SHA-1 (A534 4)	128-bit key size	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "HMAC SHA1 self test failed"	Message Authentication	Power On Crypto Library Load
HMAC-SHA2-256 (A534 4)	128-bit key size	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "HMAC SHA256 self test failed"	Message Authentication	Power On Crypto Library Load
HMAC-SHA2-512	2048-bit key size	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "HMAC SHA512 self test failed"	Message Authentication	Power On Crypto Library Load

Algori thm or Test	Test Properties	Test Metho d	Test Typ e	Indicator	Details	Condit ions
(A534 4)						
KAS- ECC- SSC Sp800 -56Ar3 (A534 4)	P-224 Curve	KAT	CAS T	Success: "Power-up self- tests succeeded" Error: "KAS ECC Primitive Z self test failed"	Shared Secret "Z" Computa tion"	Power On Crypto Library Load
KAS- FFC- SSC Sp800 -56Ar3 (A534 4)	ffdhe2048	KAT	CAS T	Success: "ffdhe_crypto_algo_self_tes t passed" Error: "assertion failed: '(left == right)'"	Shared Secret "Z" Computa tion	Power On Crypto Library Load
KDF SP800 -108 (A534 4)	256-bit key size Feedback mode KDF with HMAC-SHA2- 256	KAT	CAS T	Success: "Power-up self- tests succeeded" Error: "KDF108 self test failed"	Key Derivatio n	Power On Crypto Library Load
RSA Decry ption Primiti ve Sp800 -56Br2 (A534 4)	2048-bit key size SHA-256	KAT	CAS T	Success: "Power-up self- tests succeeded" Error: "RSA DP self test failed"	Decrypt	Power On Crypto Library Load
SP 800- 132 KDF	2 iterations 256-bit key size HMAC SHA2- 512	KAT	CAS T	Success: "Power-up self- tests succeeded" Error: "PBKDF2 with HMAC- SHA2-512 self test failed"	Key Derivatio n	Power On Crypto Library Load
RSA SigGe n	2048-bit key size, SHA2-256 PKCS1 v1.5	KAT	CAS T	Success: "Power-up self- tests succeeded" Error: "RSA self test failed"	Sign	Power On Crypto Library Load
RSA SigVer	2048-bit key size, SHA2-256 PKCS1 v1.5	KAT	CAS T	Success: "Power-up self- tests succeeded" Error: "RSA self test failed"	Verify	Power On Crypto Library Load

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (A5344)		KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "SHA1 self test failed"	Hash	Power On Crypto Library Load
SHA2-256 (A5344)		KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "SHA256 self test failed"	Hash	Power On Crypto Library Load
SHA2-512 (A5344)		KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "SHA512 self test failed"	Hash	Power On Crypto Library Load
SHA3	SHA3-224 SHA3-256 SHA3-384 SHA3-512	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "SHA3 self test failed"	Hash	Power On Crypto Library Load
TLS v1.2 KDF RFC7627 (A5344)	RFC 7627 SHA-256 SHA-384	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "TLS 1.2 KDF self test failed"	Key Derivation	Power On Crypto Library Load
TLS v1.3 KDF (A5344)	RFC 8446 SHA-256 SHA-384	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "TLS 1.3 self test failed"	Key Derivation	Power On Crypto Library Load
ENT (P)	Bit Pattern Counting using Six (6) Different Bit Patterns in Sliding Window	Developer-Defined CHT per SP 800-90B section 4.5	CAS T	Success: 0 Error: "FIPS conditional test failure: Error in cryptographic operation - RNG failed"	Entropy Gathering	Upon entropy source (RDSEED) access
DRBG	Counter DRBG with AES-256	CRNG T	CAS T	Success: 0 Error: "FIPS conditional test failure: Error in cryptographic operation - RNG failed"	Instantiate, Generate, Reseed	Upon DRBG access
EDDSA	ED-25519	PCT	PCT	Success: Ok Error: "FIPS conditional test failure:	Sign, Verify	Upon EDDSA

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				Pairwise consistency test failed. Sign / Verify test failed."		A key pair generation
ECDH and DH (PCT)	KAS-ECC, KAS-FFC	PCT	PCT	Success: Ok Error: "FipsCheckError::General(FFDHE Pairwise Consistency Test: failed)" / "FipsCheckError::Mbedtls(EcplInvalidKey)"	Key Agreement	Upon (EC)DH key pair generation and import
ECDH and DH (CAST)	KAS-ECC, KAS-FFC	Full Public-Key Validity	CAS T	Success: Ok Error: "FipsCheckError::General(FFDHE Full Public-Key Validity: failed)" / "FipsCheckError::Mbedtls(EcplInvalidKey)"	Key Agreement	Upon (EC)DH key pair generation and import
RSA (PCT)	SHA-256	PCT	PCT	Success: Ok Error: "FIPS conditional test failure: Pairwise consistency test failed. Sign / Verify test failed." / "FIPS conditional test failure: Pairwise consistency test failed. Encryption / Decryption test failed."	Sign, Verify, Encrypt, Decrypt	Upon RSA key pair generation
Firmware Load Test	ECDSA P-256 SHA2-256	Digital Signature Verification	SW/ FW Load	Success: Ok Error: "Firmware verification failed."	Verify	Upon Firmware Load
Plugin Load Test	RSA (2048 - 4096) ECDSA (P-256, P-384, or P-521)	Digital Signature	SW/ FW Load	Success: Ok Error: "failed to verify plugin code signature"	Verify	Upon plugin Load
ECDSA	SHA-256	PCT	PCT	Success: Ok Error: "FIPS conditional test failure: Pairwise consistency test failed. Sign / Verify test failed."	Sign, Verify	Upon ECDSA key pair generation
LMS and HSS KeyGen	HSS, One Level LMS_SHA256_M24_H5	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "HSS self test failed"	Sign, Verify	Power On Crypto Library Load

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
LMS SigGen	HSS, One Level LMS_SHA256_M24_H5	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "HSS self test failed"	Sign	Power On Crypto Library Load
LMS SigVer	HSS, One Level LMS_SHA256_M24_H5	KAT	CAS T	Success: "Power-up self-tests succeeded" Error: "HSS self test failed"	Verify	Power On Crypto Library Load

Table 23: Conditional Self-Tests

Per IG D.J Additional Comment #3, it is not required that the conditioning component algorithms used for the entropy source are self-tested.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Firmware Integrity Test	HMAC-SHA2-256	SW/FW Integrity	N/A	N/A
Key Store Integrity Test	16-bit Sum-of-bytes Checksum	SW/FW Integrity	N/A	N/A
Filesystem Integrity Test	32-bit CRC32C Checksum	SW/FW Integrity	N/A	N/A

Table 24: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES Encrypt	KAT	CAST	24 hours	Automatic
AES Decrypt	KAT	CAST	24 hours	Automatic
AES GCM Encrypt	KAT	CAST	24 hours	Automatic
AES GCM Decrypt	KAT	CAST	24 hours	Automatic
AES CCM Encrypt	KAT	CAST	24 hours	Automatic
AES CCM Decrypt	KAT	CAST	24 hours	Automatic
AES Key Wrap	KAT	CAST	24 hours	Automatic
AES Key Unwrap	KAT	CAST	24 hours	Automatic
AES CMAC Encrypt	KAT	CAST	24 hours	Automatic

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES CMAC Decrypt	KAT	CAST	24 hours	Automatic
AES FF1 Encrypt	KAT	CAST	24 hours	Automatic
AES FF1 Decrypt	KAT	CAST	24 hours	Automatic
Counter DRBG (A5344)	KAT	CAST	24 hours	Automatic
ECDSA SigGen (FIPS186-5) (A5344)	KAT	CAST	24 hours	Automatic
ECDSA SigVer (FIPS186-5) (A5344)	KAT	CAST	24 hours	Automatic
EDDSA SigGen (A5344)	KAT	CAST	24 hours	Automatic
EDDSA SigVer (A5344)	KAT	CAST	24 hours	Automatic
HMAC-SHA-1 (A5344)	KAT	CAST	24 hours	Automatic
HMAC-SHA2-256 (A5344)	KAT	CAST	24 hours	Automatic
HMAC-SHA2-512 (A5344)	KAT	CAST	24 hours	Automatic
KAS-ECC-SSC Sp800-56Ar3 (A5344)	KAT	CAST	24 hours	Automatic
KAS-FFC-SSC Sp800-56Ar3 (A5344)	KAT	CAST	24 hours	Automatic
KDF SP800-108 (A5344)	KAT	CAST	24 hours	Automatic
RSA Decryption Primitive Sp800-56Br2 (A5344)	KAT	CAST	24 hours	Automatic
SP 800-132 KDF	KAT	CAST	24 hours	Automatic
RSA SigGen	KAT	CAST	24 hours	Automatic
RSA SigVer	KAT	CAST	24 hours	Automatic
SHA-1 (A5344)	KAT	CAST	24 hours	Automatic
SHA2-256 (A5344)	KAT	CAST	24 hours	Automatic
SHA2-512 (A5344)	KAT	CAST	24 hours	Automatic
SHA3	KAT	CAST	24 hours	Automatic
TLS v1.2 KDF RFC7627 (A5344)	KAT	CAST	24 hours	Automatic

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
TLS v1.3 KDF (A5344)	KAT	CAST	24 hours	Automatic
ENT (P)	Developer-Defined CHT per SP 800-90B section 4.5	CAST	Every entropy source access	Automatic
DRBG	CRNGT	CAST	Every DRBG access	Automatic
EDDSA	PCT	PCT	Every EDDSA key pair generation	Automatic
ECDH and DH (PCT)	PCT	PCT	Every (EC)DH public key generation and import	Automatic
ECDH and DH (CAST)	Full Public-Key Validity	CAST	Every (EC)DH public key generation and import	Automatic
RSA (PCT)	PCT	PCT	Every RSA key pair generation	Automatic
Firmware Load Test	Digital Signature Verification	SW/FW Load	Every Firmware Load	Automatic
Plugin Load Test	Digital Signature	SW/FW Load	Every Plugin Load	Automatic
ECDSA	PCT	PCT	Every ECDSA key pair generation	Automatic
LMS and HSS KeyGen	KAT	CAST	24 hours	Automatic
LMS SigGen	KAT	CAST	24 hours	Automatic
LMS SigVer	KAT	CAST	24 hours	Automatic

Table 25: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Hard Error State	Unrecoverable errors that indicate an equipment malfunction that may require maintenance, service, or repair of the cryptographic module	Cryptographic Algorithm Self-Test (CAST) failure Integrity Test failure	Power Cycle	A status message is output to the logs indicating which self-test failed
Soft Error State	Recoverable errors that indicate a transient error, such as an operator	Firmware Load Test failure Plugin Load Test failure Public Key	Automatic	A status code is output to the operator

Name	Description	Conditions	Recovery Method	Indicator
	providing invalid inputs to a function	Checks on externally-provided public keys failure		indicating that the operation failed

Table 26: Error States

10.5 Operator Initiation of Self-Tests

The full set of self-tests can be initiated on-demand by an operator by executing the “system reboot” command via the CLI, or alternatively just the CASTs can be executed on-demand by executing “sdkms restart.”

10.6 Additional Information

The module performs power-up and conditional self-tests. Upon successful execution of all power-up self-tests, the module provides the following status:

“Software Integrity test succeeded”
“Power-up self-tests succeeded”

Upon failure of a power-up or conditional self-test, the module halts its operation and enters an error state. The preceding tables describe the self-tests implemented by the module along with the associated, individual status messages.

The term “FIPS” by itself is not a FIPS 140-3 recognized term. It is part of the module code and will be updated in a future release. The correct term is “Approved Mode”.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Crypto Officer is required to follow the vendor procedural control guidelines to set up and install the module after it is received.

Here is a summary of the procedure. For more information, please refer to the [\[User Guide\]](#), at Fortanix support portal and viewing the guides under sub-section “Getting Started with Fortanix DSM”.

Module unpacking must be done in a secure location where only authorized personnel have access.

Before installing Fortanix DSM Appliance, it must be checked by the Crypto Officer for evidence of tampering damage to the tamper-evident labels and any physical damage to the enclosure material. Fortanix DSM Appliance comes with tamper evident labels applied. If the tamper labels are damaged, then they must not be used and returned to Fortanix.

The installation must be conducted by authorized personnel who have crypto officer role in the organization. The installation must be conducted in a secure location which is accessible only by authorized personnel.

After inspection and server installation, the crypto officer needs to complete the following tasks for configuring the module as described in the user guide before the module becomes operational:

1. Log in with and change the default credentials – This step is required before continuing with the configuration.
2. Configure the network and IPMI interfaces
3. Configure DSM
4. Invoke “Get Status” Service as documented in Section 2.4 and verify the mode of operation.

11.2 Administrator Guidance

Please refer to the [\[User Guide\]](#) for the Administrator guidance. The administrator guidance can be found under the sub-section “Fortanix DSM Administration”.

11.3 Non-Administrator Guidance

Please refer to the [\[User Guide\]](#) for the Non-Administrator guidance. The non-administrator guidance can be found under the sub-sections “Fortanix DSM Administration” & “Developer Guide - Developer Operations”.

11.4 Design and Rules

1. The module enforces logical separation between all data inputs, data outputs, control inputs, and status outputs via the cryptographic module API and CLI.
2. The cryptographic module inhibits all data output during self-tests and error states. The data output interface is logically disconnected from the processes performing self-tests and zeroization.
3. Power-up self-tests do not require any operator intervention.
4. The cryptographic module does not support a maintenance interface or maintenance role.
5. The cryptographic module does not support a bypass capability.
6. Results of previous authentications are cleared when the module is powered off. The operator is required to re-authenticate into the module.
7. The operator can Power cycle the module to exit the error states and resume normal operation.
8. The module protects public keys and CSPs from unauthorized disclosure, unauthorized modification, and unauthorized substitution.
9. The module does not output intermediate key values.
10. The module complies with FIPS 140-3 IG C.H requirements for AES-GCM:
 - a. For TLS V1.2 Protocol, the module constructs the IV (internally) as allowed per Technique #1 in FIPS 140-3 IG C.H for Industry Protocols. The IV total length is 96-bits, where the fixed IV length is 32-bits and nonce_explicit part of the IV is 64-bits. The GCM key and IV are session specific; if the module loses power the implementation is required to re-initialize a TLS V1.2 session, creating a new IV altogether.
 - b. For TLS V1.3 Protocol, the module constructs the IV (internally) as allowed per Technique #5 in FIPS 140-3 IG C.H for Industry Protocols. The IV total length is 96-bits, where a 64-bit record sequence number is left-padded and XORed with a

mutually derived IV (as specific in RFC 8446). The mutually derived IV is derived from a secret that is originally based on 32 bytes of random data generated by the module. This assures that the odds of a collision are less than $2^{(-32)}$. The GCM key and IV are session specific; if the module loses power the implementation is required to re-initialize a TLS V1.3 session, creating a new IV altogether.

- c. For the Encrypt/Decrypt service, a 96-bit IV is constructed from the output of the CTR_DRBG, allowed as per Technique #2 in FIPS 140-3 IG C.H for IVs generated “internally at its entirety randomly”. In case the module’s power is lost and then restored, a new key for use with the AES GCM encryption/decryption will be generated from the output of the CTR_DRBG.
- 11. In accordance with FIPS 140-3 IG D.H, the cryptographic module performs Cryptographic Key Generation (CKG) as per SP 800-133r2 (Vendor Affirmed). The resulting generated symmetric key and/or generated seed for asymmetric key generation, are from the unmodified output of the SP 800-90A DRBG.
- 12. The module complies with FIPS 140-3 IG C.J requirements for SP 800-38G. The module supports radix values of 2 to 64, min length is based on radix such that $\text{radix}^{\text{minlen}} \geq 1,000,000$, max length is 2^{16} and Max tweak length (maxTlen) is same as maxlen.

11.6 End of Life

To securely sanitize the module at end of life, an operator must exercise the module’s zeroization service via the module’s CLI. Furthermore, if secure destruction is desired then the operator must first securely sanitize the module and then proceed to physically dispose of the module appliance using their preferred electronics waste disposal facility.

12 Mitigation of Other Attacks

Not Applicable - Fortanix does not claim any mitigation strategies for the Fortanix DSM Appliance.