Cisco Systems, Inc.

Linux Kernel FIPS Object Module (KFOM) Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This is Cisco Systems, Inc.'s non-proprietary security policy for Linux Kernel FIPS Object Module (KFOM) Cryptographic Module (hereinafter referred to as KFOM or Module) firmware version 2.0.5. The following details how this module meets the security requirements of FIPS 140-3, SP 800-140 and ISO/IEC 19790 for a Security Level 1 Firmware hybrid cryptographic module.

The security requirements cover areas related to the design and implementation of a cryptographic module. These areas include cryptographic module specification; cryptographic module interfaces; roles, services, and authentication; software/firmware security; operational environment; physical security; non-invasive security; sensitive security parameter management; self-tests; life-cycle assurance; and mitigation of other attacks. The following table indicates the actual security levels for each area of the cryptographic module.

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	1
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	1
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Cisco Linux Kernel FIPS Object Module (KFOM) Cryptographic Module is a firmware hybrid cryptographic library in a multi-chip standalone embodiment that allows for Linux kernel applications to use approved algorithms. It does not implement any security protocols. Instead, it only provides Linux kernel applications access to approved algorithms.

The module is intended to run on the UCS C220 M5 host platform or any general-purpose computer, so the physical perimeter of the module is the tested platforms. The cryptographic module comprises the Cisco Linux Kernel FIPS Object Module (KFOM) Cryptographic Module (Firmware Version: 2.0.5) which is a kernel object file linux_kfom_2_0_5.ko and the processors (only for algorithm acceleration) which only operates in the approved mode of operation which is set at manufacture. The module is validated according to FIPS 140-3 at overall security level 1. Please refer to Table 1 above for the individual areas.

Cisco UCS C220 M5 unifies computing, networking, management, virtualization, and storage access into a single integrated architecture, enabling end-to-end server visibility, management, and control in both bare metal and virtualized environments.

The module has been tested on the following Operational Environments.

Module Type: Firmware-hybrid

Module Embodiment: Multi-Chip Standalone

Module Characteristics:

Cryptographic Boundary:

The KFOM cryptographic module (red box) is a non-modifiable, multi-chip standalone firmware hybrid cryptographic module providing cryptographic support to the Kernel which takes data in and out from the host application via the API (libkcapi) feeding into the kernel. All processing is done on the listed processors in Table 2 above. The KFOM performs no communications other than with the consuming application. The block diagram below shows the boundary of the Tested Operational Environment's Physical Perimeter (TOEPP) being defined as the physical perimeter of the tested platform enclosure around which everything runs. Then the cryptographic boundary is the KFOM (red box) and its interfaces with the operational environment.

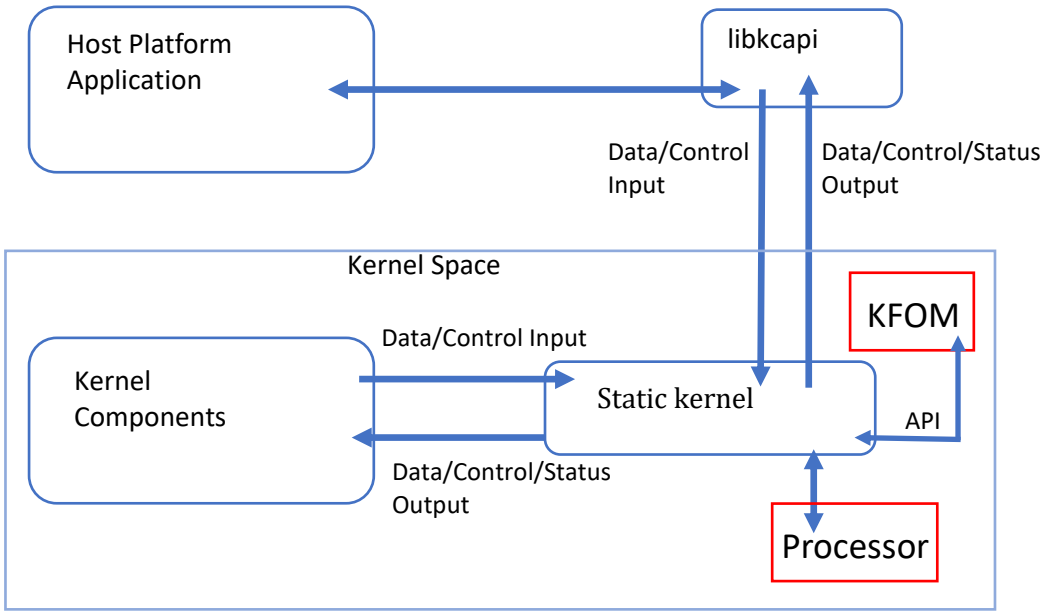

Figure 1: Block Diagram

Note: The kernel object file linux_kfom_2_0_5.ko is represented by KFOM (red box) in the Block Diagram.

The hybrid disjoint hardware component, Intel Xeon Platinum 8180 (Skylake), is shown below with approximate dimensions of 76.0mm x 56.5mm.

Figure 2: Intel Xeon Platinum 8180 (Skylake)

Tested Operational Environment's Physical Perimeter (TOEPP):

The boundary of the TOEPP is defined as the entire chassis unit's physical perimeter encompassing the "top," "front," "left," "right," "rear" and "bottom" surfaces of the case, and shown in the figures below.

Figure 3 - UCS C220 M5 Front View

Figure 4 - UCS C220 M5 Rear View

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

N/A for this module.

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

Package or File Name	Software/ Firmware Version	Features	Integrity Test
linux_kfom_2_0_5.ko	2.0.5		HMAC-SHA2-512

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Tested Module Identification – Hybrid Disjoint Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
Intel Xeon Platinum 8180 (Skylake)	Intel Xeon Platinum 8180 (Skylake)		Intel Xeon Platinum 8180 (Skylake)	

Table 3: Tested Module Identification – Hybrid Disjoint Hardware

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Ubuntu 22.04	UCS C220 M5	Intel Xeon Platinum 8180 (Skylake)	Yes		2.0.4

Table 4: Tested Operational Environments - Software, Firmware, Hybrid

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform
Linux 5.4-6.12	MX67
Linux 5.4-6.12	MX68
Linux 5.4-6.12	MX75
Linux 5.4-6.12	MX85
Linux 5.4-6.12	MX95
Linux 5.4-6.12	MX105
Linux 5.4-6.12	MX250
Linux 5.4-6.12	MX450
Linux 5.4-6.12	MX68CW
Linux 5.4-6.12	C8455-G2
Linux 5.4-6.12	MG52
Linux 5.4-6.12	MR28
Linux 5.4-6.12	MR36
Linux 5.4-6.12	MR44
Linux 5.4-6.12	MR56
Linux 5.4-6.12	MR57
Linux 5.4-6.12	MR76
Linux 5.4-6.12	MR78
Linux 5.4-6.12	MR86
Linux 5.4-6.12	CW9166I
Linux 5.4-6.12	CW9166ID1
Linux 5.4-6.12	CW9164I

Operating System	Hardware Platform
Linux 5.4-6.12	CW9162I
Linux 5.4-6.12	CW9163E
Linux 5.4-6.12	CW9178I
Linux 5.4-6.12	CW9176I
Linux 5.4-6.12	CW9176D1

Table 5: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

2.3 Excluded Components

N/A for this module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved Mode of Operation	The module is always in the approved mode of operation.	Approved	The successful completion of a service is an implicit indicator for the use of an approved service.

Table 6: Modes List and Description

By design, the module is only able to support approved mode of operation. The module doesn't claim the implementation of a degraded mode operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A6627	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS3	A6627	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A6627	Key Length - 128, 192, 256	SP 800-38C
AES-CMAC	A6627	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A6627	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A6627	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-GCM	A6627	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
AES-XTS Testing Revision 2.0	A6627	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A6627	Prediction Resistance - Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA SigVer (FIPS186-5)	A6627	Curve - P-256, P-384 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-5
Hash DRBG	A6627	Prediction Resistance - Yes Mode - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A6627	Prediction Resistance - Yes Mode - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A6627	Key Length - Key Length: 256-448 Increment 8	FIPS 198-1
HMAC-SHA2-224	A6627	Key Length - Key Length: 256-448 Increment 8	FIPS 198-1
HMAC-SHA2-256	A6627	Key Length - Key Length: 256-448 Increment 8	FIPS 198-1
HMAC-SHA2-384	A6627	Key Length - Key Length: 256-448 Increment 8	FIPS 198-1
HMAC-SHA2-512	A6627	Key Length - Key Length: 256-448 Increment 8	FIPS 198-1
KAS-ECC CDH-Component SP800-56Ar3 (CVL)	A6627	Curve - P-256, P-384	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A6627	Domain Parameter Generation Methods - P-256, P-384 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
LMS SigVer	A6627	LMS Modes - LMS_SHA256_M24_H10, LMS_SHA256_M24_H15, LMS_SHA256_M24_H20, LMS_SHA256_M24_H25, LMS_SHA256_M24_H5, LMS_SHA256_M32_H10, LMS_SHA256_M32_H15, LMS_SHA256_M32_H20, LMS_SHA256_M32_H25, LMS_SHA256_M32_H5	SP 800-208
SHA-1	A6627	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-224	A6627	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-256	A6627	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-384	A6627	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-512	A6627	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA3-224	A6627	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHA3-256	A6627	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHA3-384	A6627	Message Length - Message Length: 0-65528 Increment 8	FIPS 202
SHA3-512	A6627	Message Length - Message Length: 0-65528 Increment 8	FIPS 202

Table 7: Approved Algorithms

Vendor-Affirmed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
ECDSA SigVer	DigSig-SigVer	ECDSA Signature Verification Security Function		ECDSA SigVer (FIPS186-5): (A6627)

Name	Type	Description	Properties	Algorithms
LMS SigVer	DigSig-SigVer	LMS Signature Verification Security Function		LMS SigVer: (A6627)
Encryption/Decryption	BC-Auth BC-UnAuth	Encryption/Decryption Security Function		AES-CBC: (A6627) AES-CBC-CS3: (A6627) AES-CCM: (A6627) AES-CMAC: (A6627) AES-CTR: (A6627) AES-ECB: (A6627) AES-GCM: (A6627) AES-XTS Testing Revision 2.0: (A6627)
Keyed Hash	MAC	Authenticated Hash Security Function		HMAC-SHA-1: (A6627) HMAC-SHA2-224: (A6627) HMAC-SHA2-256: (A6627) HMAC-SHA2-384: (A6627) HMAC-SHA2-512: (A6627) SHA-1: (A6627) SHA2-224: (A6627) SHA2-256: (A6627) SHA2-384: (A6627) SHA2-512: (A6627)
Message Digest	SHA	Unauthenticated Hash Security Function		SHA-1: (A6627) SHA2-224:

Name	Type	Description	Properties	Algorithms
				(A6627) SHA2-256: (A6627) SHA2-384: (A6627) SHA2-512: (A6627) SHA3-224: (A6627) SHA3-256: (A6627) SHA3-384: (A6627) SHA3-512: (A6627)
DRBG (AES_CTR, HMAC, or Hash)	DRBG	DRBG Security Function		Hash DRBG: (A6627) HMAC DRBG: (A6627) Counter DRBG: (A6627)
KAS-ECC Shared Secret Computation	KAS-SSC	KAS-ECC Shared Secret Computation Security Function	IG:IG D.F Scenario (2) path (1)	KAS-ECC-SSC Sp800-56Ar3: (A6627) KAS-ECC CDH-Component SP800-56Ar3: (A6627)

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

Algorithm Cert. #A6627 was tested for the OE with PAA.

AES-GCM

Use of external IV with GCM is exclusively permitted for decryption operations or where the GCM is used to support the protocol specific implementation used to protect connections to the calling applications. The GCM IV is generated per FIPS 140-3 IG C.H technique #3. The counter portion of the IV is set by the module within its cryptographic boundary. When the IV exhausts the maximum number of possible values for a given session key, the first party, client or server, to encounter this condition will trigger a handshake to establish a new encryption key.

In case the module's power is lost and then restored, a new key for use with the AES GCM encryption/decryption shall be established.

AES-XTS
The length of a single data unit encrypted or decrypted with AES XTS shall not exceed 2^20 AES blocks of data per XTS instance. An XTS instance is defined in Section 4 of SP 800-38E.

The XTS mode shall only be used for the cryptographic protection of data on storage devices. It shall not be used for other purposes, such as the encryption of data in transit. To meet the requirement stated in IG C.I, AES-XTS keys (i.e., Key_1 and Key_2) entered into the module shall be generated and/or established independently according to NIST SP 800-133rev2, Section 6.3. for an approved use of AES-XTS. Additionally, the module implements a check to ensure that the two AES keys used in AES XTS mode are not identical.

KAS-ECC CDH Component (CVL)
The KAS-ECC CDH Component (CVL) is only used within the context of SP-800 56Ar3 KAS.

SHA-1
The module includes an implementation of SHA-1 for hashing. The module does not implement SHA-1 in the context of signature generation or verification. This implementation will be non-Approved for all uses starting January 1, 2031. User should move to SHA2, which is available in this module.

2.8 RBG and Entropy

N/A for this module.

N/A for this module.

2.9 Key Generation

N/A for this module.

2.10 Key Establishment

KAS-ECC Shared Secret Computation:

- The module provides KAS-ECC Shared Secret Computation and KAS-ECC CDH Component compliant with SP800-56Arev3. The shared secret computation uses curves P-256 and P-384 providing between 128 and 192 bits of encryption strength.

2.11 Industry Protocols

N/A for this module.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
Input registers [Hardware component only]	Data Input	Data to be encrypted, decrypted or hashed; Keys to be used in cryptographic services.
Output registers [Hardware component only]	Data Output	Data that has been encrypted or decrypted
Control registers [Hardware component only]	Control Input	Instructions to invoke the PAA operation
Status registers [Hardware component only]	Status Output	Return Values
N/A [Firmware component only]	Data Input	Arguments for an API call that provide the data to be used or processed by the module: (1) Data to be encrypted, decrypted or hashed; (2) Keys to be used in cryptographic services; (3) Random seed material for the module's DRBG
N/A [Firmware component only]	Data Output	Arguments output from an API call: (1) Data that has been encrypted or decrypted; (2) Hashes
N/A [Firmware component only]	Control Input	Arguments for an API call used to control and configure module operation: (1) Modes, key sizes, etc. used with cryptographic services
N/A [Firmware component only]	Status Output	(1) Return values; (2) Status information regarding the module; (3) Status information regarding the invoked service/operation
N/A	Control Output	N/A

Table 9: Ports and Interfaces

The module's physical perimeter encompasses the case of the tested platform mentioned in Table 2. The module provides its logical interfaces via Application Programming Interface (API) calls. The logical interfaces provided by the module are mapped onto the FIPS 140-3 interfaces (data input, data output, control input, control output and status output) as shown above. The module's data output interface will be disabled when performing the self-test service, zeroization service, or when in an error state.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module supports Crypto Officer (CO) role. The cryptographic module does not provide any authentication methods. The module does not allow concurrent operators. The Crypto Officer is implicitly assumed based on the service requested.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 10: Roles

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show Status	Provide Module's current status (return codes and/or syslog messages)	N/A	API Call used to show Module's Status	Module's operational status	None	Crypto Officer
Show Version	Provide Module's name and version information	N/A	API Call to show version	Module's ID and versioning information	None	Crypto Officer
Perform Self-Tests	Perform self-tests (pre-operational and conditional self-tests)	N/A	Power Cycling the Module	Status of the self-test results	None	Crypto Officer Unauthenticated
Perform Symmetric Encryption	Perform Encryption with AES algorithm	The successful completion of a service is an implicit	API Call to trigger symmetric encryption	Status of AES encryption.	Encryption/Decryption	Crypto Officer - AES Key: R,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		indicator for the use of an approved service.				
Perform Symmetric Decryption	Perform Decryption with AES algorithm	The successful completion of a service is an implicit indicator for the use of an approved service.	API Call to trigger symmetric decryption	Status of AES decryption	Encryption/Decryption	Crypto Officer - AES Key: R,E
Generate Keyed Hash	Generate Keyed Hash with HMAC	The successful completion of a service is an implicit indicator for the use of an approved service	API Call to trigger generation of authenticated hash	Status of HMAC	Keyed Hash	Crypto Officer - Authentication Key: R,E
Generate Message Digest	Generate Hash with SHA	The successful completion of a service is an implicit indicator for the use of	API Call to trigger generation of message digest	Status of SHA	Message Digest	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		an approved service				
Configure Random Number Generation	Configure DRBG usage	The successful completion of a service is an implicit indicator for the use of an approved service	API Call to set DRBG	Status of DRBG	DRBG (AES_CTR, HMAC, or Hash)	Crypto Officer - DRBG Entropy Input: G,W,E - DRBG Seed: G,W,E - DRBG Internal State (V, Key): G,W,E - DRBG Internal State (V, C): G,W,E
Perform Zeroization	Perform Zeroization	The successful completion of a service is an implicit indicator for the use of an approved service	Removal of Power	Status of Zeroization	None	Crypto Officer - DRBG Entropy Input: Z - DRBG Seed: Z - DRBG Internal State (V, Key): Z - DRBG Internal State (V, C): Z - ECDSA Public Key: Z - AES Key: Z - Authentication Key: Z - LMS Public Key: Z
Verify LMS Signature	LMS digital signature verification	The successful completion of a service	API Call to perform LMS signature verification	Status of LMS Sig Ver	LMS SigVer	Crypto Officer - LMS Public Key: R,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		is an implicit indicator for the use of an approved service				
Verify ECDSA Signature	ECDSA digital signature verification	The successful completion of a service is an implicit indicator for the use of an approved service	API Call to perform ECDSA signature verification	Status of ECDSA SigVer	ECDSA SigVer	Crypto Officer - ECDSA Public Key: R,E
Compute KAS-ECC Shared Secret	KAS-ECC shared secret computation	The successful completion of a service is an implicit indicator for the use of an approved service	API Call to perform KAS-ECC shared secret computation	Status of KAS-ECC shared secret computation	KAS-ECC Shared Secret Computation	Crypto Officer - Party A ECDH Private Key: R,E - Party B ECDH Public Key: R,E - ECDH Shared Secret: R,E

Table 11: Approved Services

4.4 Non-Approved Services

N/A for this module.

4.5 External Software/Firmware Loaded

N/A for this module.

4.6 Additional Information

The module supports unauthenticated service. The unauthenticated operator can trigger the self-test service by power-cycling the module.

5 Software/Firmware Security

5.1 Integrity Techniques

The Module is provided in the form of binary executable code. To ensure firmware security, the Module is protected by HMAC-SHA2-512 (Algo Cert. #A6627) algorithm. At Module's initialization, the integrity of the runtime executable is verified using a HMAC-SHA2-512 digest which is compared to a value computed at build time. If at the load time the MAC does not match the stored, known MAC value, the Module would enter an Error state with all crypto functionality inhibited.

5.2 Initiate on Demand

The integrity test is performed as part of the Pre-Operational Self-Tests. It is automatically executed at power-on. The operator can initiate the integrity test on demand by power cycling the host platform.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Non-Modifiable

How Requirements are Satisfied:

The module is a firmware hybrid module, which is operated in a non-modifiable operational environment per FIPS 140-3 level 1 specifications. The module's firmware version running on the tested platform is 2.0.5.

The module has control over its own SSPs. The process and memory management functionality of the host device's OS prevent unauthorized access to plaintext private and secret keys, intermediate key generation values and other SSPs by external processes during module execution. The module only allows access to SSPs through its well-defined API. The operational environments provide the capability to separate individual application processes from each other by preventing uncontrolled access to CSPs and uncontrolled modifications of SSPs regardless of whether this data is in the process memory or stored on persistent storage within the operational environment. Processes that are spawned by the module are owned by the module and are not owned by external processes/operators.

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Production-grade components with standard passivation	N/A	N/A

Table 12: Mechanisms and Actions Required

The hardware component of the firmware-hybrid module is made of production-grade components with standard passivation.

8 Non-Invasive Security

N/A for this module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM of the TOEPP	Volatile memory located outside of the cryptographic boundary but within the Tested Operational Environment Physical Perimeter (TOEPP)	Dynamic

Table 13: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API Call: SSP Input	Operator calling application (TOEPP)	Module	Plaintext	Manual	Electronic	
API Call: SSP Output	Module	Operator calling application (TOEPP)	Plaintext	Manual	Electronic	

Table 14: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Power off	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed.	Removing power from the module

Table 15: SSP Zeroization Methods

Power-cycling the host device will implicitly zeroize all SSPs.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG Entropy Input	Used to seed the DRBG	at least 256 bits - at least 256 bits	Entropy Input - CSP			Counter DRBG (A6627) HMAC DRBG (A6627) Hash DRBG (A6627)
DRBG Seed	Used in DRBG Generation	256 bits - 256 bits	DRBG Seed - CSP			Counter DRBG (A6627) HMAC DRBG (A6627) Hash DRBG (A6627)
DRBG Internal State (V, Key)	Used in DRBG Generation	256 bits - 256 bits	DRBG Internal State - CSP			Counter DRBG (A6627) HMAC DRBG (A6627)
DRBG Internal State (V, C)	Used in DRBG Generation	256 bits - 256 bits	DRBG Internal State - CSP			Hash DRBG (A6627)
ECDSA Public Key	Used for verifying ECDSA signatures	Curves: 256, 384 - 128 to 192 bits	Public Key - PSP			ECDSA SigVer
AES Key	Used for encryption and decryption services.	128, 192, 256 bits - 128, 192, 256 bits	Symmetric Key - CSP			Encryption/Decryption
Authentication Key	Key used in Keyed Hashing	160-512 bits	Integrity Key - CSP			Keyed Hash

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	for integrity assurance	- 160-512 bits				
LMS Public Key	Used for verifying LMS signatures	256 - 256	Public Key - PSP			LMS SigVer
Party A ECDH Private Key	Used to derive the ECDH Shared Secret	Curves: 256 and 384 - 128-192 bits	Private Key - CSP			KAS-ECC Shared Secret Computation
Party B ECDH Public Key	Used to derive the ECDH Shared Secret	Curves: 256 and 384 - 128-192 bits	Public Key - PSP			KAS-ECC Shared Secret Computation
ECDH Shared Secret	Established by the module and returned to the calling application	Curves: 256 and 384 - 128-192 bits	Shared Secret - CSP		KAS-ECC Shared Secret Computation	

Table 16: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DRBG Entropy Input		RAM of the TOEPP:Plaintext		Power off	
DRBG Seed		RAM of the TOEPP:Plaintext		Power off	
DRBG Internal State (V, Key)		RAM of the TOEPP:Plaintext		Power off	
DRBG Internal State (V, C)		RAM of the TOEPP:Plaintext		Power off	
ECDSA Public Key	API Call: SSP Input	RAM of the TOEPP:Plaintext		Power off	
AES Key	API Call: SSP Input	RAM of the TOEPP:Plaintext		Power off	
Authentication Key	API Call: SSP Input	RAM of the TOEPP:Plaintext		Power off	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
LMS Public Key	API Call: SSP Input	RAM of the TOEPP:Plaintext		Power off	
Party A ECDH Private Key	API Call: SSP Input	RAM of the TOEPP:Plaintext		Power off	
Party B ECDH Public Key	API Call: SSP Input	RAM of the TOEPP:Plaintext		Power off	
ECDH Shared Secret	API Call: SSP Output	RAM of the TOEPP:Plaintext		Power off	

Table 17: SSP Table 2

The cryptographic module is passed a pointer to the cryptographic keys as API parameters, associated by memory location. The application calling the cryptographic module passes keys in plaintext within the physical perimeter. The module does not perform storage of keys. All SSPs can be zeroized by power cycling the host.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-512 (A6627)	HMAC-SHA2-512	KAT	SW/FW Integrity	Module is in normal state	Firmware Integrity Test

Table 18: Pre-Operational Self-Tests

The pre-operational firmware integrity tests along with all of the Conditional Self-tests are performed automatically when the module is powered on, before the module transitions into the operational state. Prior to the running of HMAC-SHA2-512 integrity test, the module will run and must pass the HMAC-SHA2-512 conditional Known Answer Test (KAT). While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed. The module transitions to the operational state only after all self-tests are passed successfully. A successful log message is provided by the module after the successful completion of the self-tests.

If an error occurs to a valid approved algorithm during the self-test, the module enters hard error state, and the Linux kernel will print an error message to the console and data output from the data output interface is inhibited. This results in the system shutting down. The error state can only be cleared by reloading the module. All self-tests must be completed successfully before the module transitions to the operational state.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB Encrypt KAT (A6627)	256 bits	KAT	CAST	Module is in normal state	Encrypt KAT	Power Up
AES-ECB Decrypt KAT (A6627)	256 bits	KAT	CAST	Module is in normal state	Decrypt KAT	Power Up
AES-GCM Authenticated Encrypt KAT (A6627)	256 bits	KAT	CAST	Module is in normal state	Encrypt KAT	Power Up
AES-GCM Authenticated Decrypt KAT (A6627)	256 bits	KAT	CAST	Modue is in normal state	Decrypt KAT	Power Up
Counter DRBG Instantiate KAT (A6627)	AES-128	KAT	CAST	Module is in normal state	Instantiate KAT	Power Up
Counter DRBG Generate KAT (A6627)	AES-128	KAT	CAST	Module is in normal state	Generate KAT	Power Up
Counter DRBG Reseed KAT (A6627)	AES-128	KAT	CAST	Module is in normal state	Reseed KAT	Power Up
Hash DRBG Instantiate KAT (A6627)	SHA-256	KAT	CAST	Module is in normal state	Instantiate KAT	Power Up
Hash DRBG Generate KAT (A6627)	SHA-256	KAT	CAST	Module is in normal state	Generate KAT	Power Up
Hash DRBG Reseed KAT (A6627)	SHA-256	KAT	CAST	Module is in normal state	Reseed KAT	Power Up
HMAC DRBG Instantiate KAT (A6627)	HMAC-SHA2-256	KAT	CAST	Module is in normal state	Instantiate KAT	Power Up
HMAC DRBG Generate KAT (A6627)	HMAC-SHA2-256	KAT	CAST	Module is in normal state	Generate KAT	Power Up
HMAC DRBG Reseed KAT (A6627)	HMAC-SHA2-256	KAT	CAST	Module is in normal state	Reseed KAT	Power Up
HMAC-SHA2-512 KAT (A6627)	SHA2-512	KAT	CAST	Module is in normal state	HMAC-SHA2-512	Power Up
SHA3-256 KAT (A6627)	SHA3-256	KAT	CAST	Module is in normal state	SHA3-256	Power Up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
LMS SigVer KAT (A6627)	SHA2-256	KAT	CAST	Module is in normal state	LMS SHA2-256	Power Up
ECDSA SigVer (FIPS186-5) KAT (A6627)	Curve: P-256 with SHA2-256	KAT	CAST	Module is in normal state	ECDSA SigVer	Power Up
KAS-ECC-SSC Sp800-56Ar3 KAT (A6627)	P-256 Curve	KAT	CAST	Module is in normal state	Primitive Z KAT	Power Up

Table 19: Conditional Self-Tests

The module performs on-demand self-tests initiated by the operator, by powering off and powering the module back on. The full suite of self-tests is then executed. The same procedure may be employed by the operator to perform periodic self-tests.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-512 (A6627)	KAT	SW/FW Integrity	Recommend every 60 Days	Reboot

Table 20: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB Encrypt KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
AES-ECB Decrypt KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
AES-GCM Authenticated Encrypt KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
AES-GCM Authenticated Decrypt KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
Counter DRBG Instantiate KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
Counter DRBG Generate KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Counter DRBG Reseed KAT (A6627)	KAT	CAST	Recommend every 60 Days	Reboot
Hash DRBG Instantiate KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
Hash DRBG Generate KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
Hash DRBG Reseed KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
HMAC DRBG Instantiate KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
HMAC DRBG Generate KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
HMAC DRBG Reseed KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
HMAC-SHA2-512 KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
SHA3-256 KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
LMS SigVer KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
ECDSA SigVer (FIPS186-5) KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot
KAS-ECC-SSC Sp800-56Ar3 KAT (A6627)	KAT	CAST	Recommend every 60 days	Reboot

Table 21: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error State	If the self-test tests fail, the module is put into an error state.	Self-test failure	Reboot the module.	System Halt.

Table 22: Error States

The error state represents an unrecoverable error. The module will automatically reload the image. In the Error State, no cryptographic services are provided, and data output is prohibited.

If the POSTs should fail, the module will transition to the *Error* state and remain until error is corrected.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The validated cryptographic module linux_kfom_2_0_5.ko was installed onto the respective test platforms listed in Table 2 above.

The tested operating systems segregate user processes into separate process spaces. Each process space is an independent virtual memory area that is logically separated from all other processes by the operating system firmware and hardware. The module functions entirely within the process space of the process that invokes it, and thus the module runs on a single user mode of operation.

The module is loaded into the tested operational environments prior to shipping. The operator only needs to power-on the platform, and the module will be available in the approved mode for use.

The operator can verify that the module is in approved mode by executing the “Show Version” service command. If the module outputs name and version, it can be considered that the module is running in approved mode.

Additional guidance document can be obtained by contacting Cisco Systems. Inc.

11.2 Administrator Guidance

Cisco provides guidance documents for their internal engineers to assist with building this module into the host platform. This includes a readme file describing the building process and other details of the module. These guidance documents are not published or publicly available as configuration of the module within the host platform is only performed by Cisco prior to sale.

11.3 Non-Administrator Guidance

N/A for this module.

12 Mitigation of Other Attacks

N/A for this module.