Cisco Systems, Inc.

FIPS 140-2 Non-Proprietary Security Policy

Cisco Systems Kernel Crypto API Cryptographic Module

Software Version 3.10

Document Version 1.5

Abstract

This document provides a non-proprietary FIPS 140-2 Security Policy for the Cisco Systems Kernel Crypto API Cryptographic Module.

Contents

Abstract 2
1. Cryptographic Module Specification 4
1.1. Module Overview 4
1.2. FIPS 140-2 validation 5
1.3. Modes of Operations 5
2. Cryptographic Module Ports and Interfaces 5
3. Roles, Services and Authentication 7
3.1. Roles 7
3.2. Services 7
3.3. Authentication 8
4. Physical Security 9
5. Operational Environment 10
5.1. Applicability 10
5.2. Policy 10
6. Cryptographic Key Management 11
6.1. CSPs/Keys 11
6.2. Key / Critical Security Parameter (CSP) Access 11
6.3. Key / CSP Storage 11
6.4. Key / CSP Zeroization 11
7. EMI/EMC 12
8. Self-Tests 13
8.1. Power-Up Self-Tests 13
8.1.1. Integrity Tests 13
8.1.2. Known Answer Tests 13
8.1.3. Conditional Tests 13
9. Guidance 14
9.1. Cryptographic Officer Guidance 14
9.1.1. Secure Installation and Startup 14
9.2. User Guidance 14
9.3. Handling Self-Test Errors 14
10. Acronyms 15

1. Cryptographic Module Specification

1.1. Module Overview

The Cisco Systems Kernel Crypto API Cryptographic Module (hereafter referred to as the "Module") is a software only cryptographic module that provides general-purpose cryptographic services to the remainder of the Linux kernel. The Cisco Systems Kernel Crypto API Cryptographic Module is software only, security level 1 cryptographic module, running on a multi-chip standalone platform.

The module is implemented as a set of shared libraries / binary files.

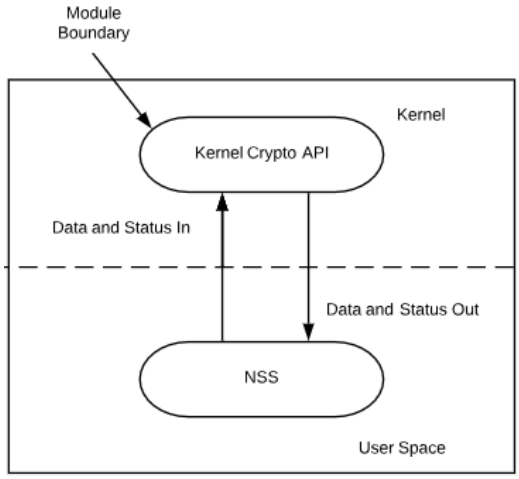

Figure 1 - Logical Boundary

The module is aimed to run on a general purpose computer; the physical boundary is the surface of the case of the target platform, as shown in the diagram below:

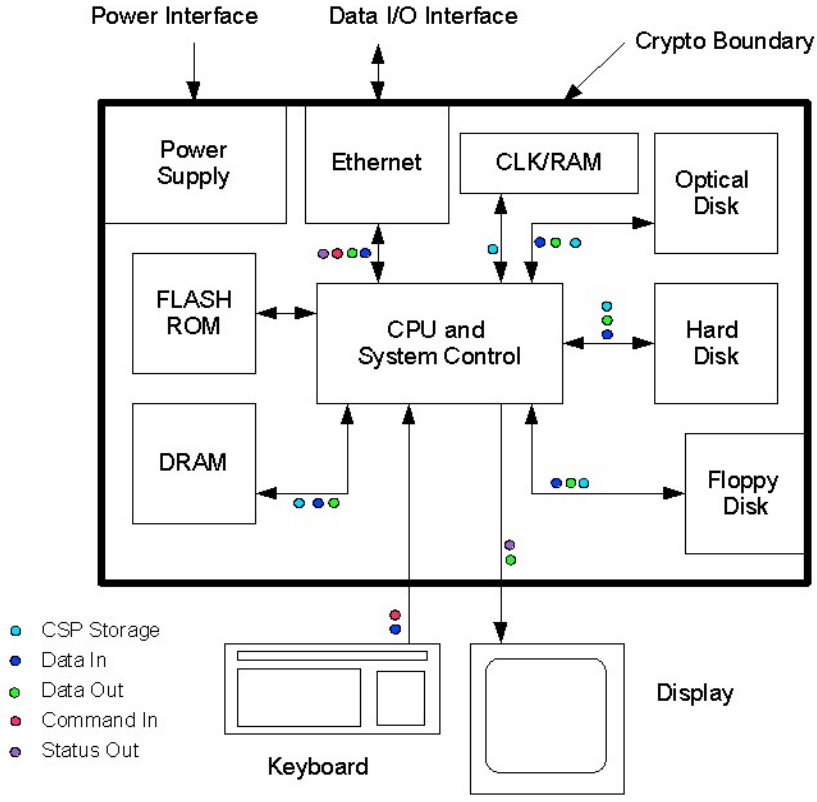

Figure 2 - Physical Boundary

The list of components required for the module to operate are defined below:

- Cisco Systems Kernel Crypto API Cryptographic Module (version 3.10)
- The module instantiation is provided by the dracut-fips (version 033-502)
- The bound module Cisco Systems NSS Module with FIPS 140-2 Certificate #3554 (hereafter referred to as the “NSS bound module” or “NSS module”)

The files comprising the module are the following:

- kernel loadable components /lib/modules/$(uname -r)/kernel/crypto/*.ko
- kernel loadable components /lib/modules/$(uname -r)/kernel/arch/x86/crypto/*.ko
- static kernel binary /boot/vmlinuz-$(uname -r)
- sha512hmac binary file for performing the integrity checks

The NSS bound module provides the HMAC-SHA-512 algorithm used by the sha512hmac binary file to verify the module integrity.

1.2. FIPS 140-2 validation

For the purpose of the FIPS 140-2 validation, the module is a software-only, multi-chip standalone cryptographic module validated at security level 1. The table below shows the security level claimed for each of the eleven sections that comprise the FIPS 140-2 standard:

FIPS 140-2 Section Title	Validation Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
Electromagnetic Interference / Electromagnetic Compatibility	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Table 1 - FIPS 140-2 Sections

The module has been tested on the following platforms with the following configuration:

Hardware Platform	Processor	Operating System
Cisco UCS M4	Intel Xeon E5-2600	CentOS Linux 7.4
Cisco UCS M5	Intel Xeon Bronze	CentOS Linux 7.4

Table 2 – Tested Platforms

The physical boundary is the surface of the case of the target platform. The logical boundary is depicted in Figure 1.

1.3. Modes of Operations

The module supports two modes of operation: the FIPS approved and non-approved modes.

Section 9.1.1 describes the Secure Installation and Startup to correctly install and configure the module. The module turns to FIPS approved mode after correct initialization, successful completion of power-on self-tests.

Invoking a non-Approved algorithm or a non-Approved key size with an Approved algorithm as listed in Table 6 will result in the module implicitly entering the non-FIPS mode of operation.

The approved services available in FIPS mode can be found in section 3.2, Table 4.

The non-approved services not available in FIPS mode can be found in section 3.2, Table 6.

2. Cryptographic Module Ports and Interfaces

As a software-only module, the module does not have physical ports. For the purpose of the FIPS 140-2 validation, the physical ports are interpreted to be the physical ports of the hardware platform on which it runs.

The logical interfaces are the application program interface (API) through which applications request services. The following table summarizes the four logical interfaces:

Logical interfaces	Description	Physical ports mapping the logical interfaces
Control In	API function calls, kernel command line	Keyboard
Status Out	API return codes, kernel logs	Display
Data In	API input parameters	Keyboard
Data Out	API output parameters	Display
Power Input	PC Power Port	Physical Power Connector

Table 3 – Ports and Logical Interface

3. Roles, Services and Authentication

3.1. Roles

The module supports the following roles:

- **User role**: performs symmetric encryption/decryption, keyed hash, message digest, random number generation, show status
- **Crypto Officer role**: performs the module installation and configuration, module's initialization, self-tests, zeroization and signature verification

The User and Crypto Officer roles are implicitly assumed by the entity accessing the module services.

3.2. Services

The module supports services available to users in the available roles. All services are described in detail in the user documentation.

The following table shows the available services, the roles allowed, the Critical Security Parameters involved and how they are accessed in the FIPS mode. 'R' stands for Read permission, 'W' stands for write permission and 'EX' stands for executable permission of the module:

Service	Algorithms	Note(s) / Mode(s)	Bounded Module	Role	CSPs	Access
Symmetric encryption/ decryption	AES	Mode: AES-CBC Operation: E/D Key Length: 128-bit, 192-bit, 256-bit	N/A	User	AES keys (128, 192, 256 bits)	R, W, EX
		Mode: AES-ECB Operation: E/D Key Length: 128-bit, 192-bit, 256-bit				
Keyed hash (HMAC)	HMAC SHA-1	MAC: 80, 96, 128, 160 BS < KS, KS = BS, KS > BS	N/A	User	HMAC keys (112 bits +)	R, W, EX
	HMAC SHA-256	MAC: 128, 192, 256 BS < KS, KS = BS, KS > BS				
	HMAC SHA-384	MAC: 128, 192, 256, 320, 384 BS < KS, KS = BS, KS > BS				
	HMAC SHA-512	MAC: 256, 320, 384, 448, 512 BS < KS, KS = BS, KS > BS				
Message digest (SHS)	SHA-1, SHA-256 SHA-384 SHA-512	N/A	N/A	User	N/A	R, W, EX
Authenticated encryption	AES CBC mode and HMAC-SHA-1, HMACSHA-256, HMACSHA-512	Encrypt-then-MAC Authenticated encryption using both AES and HMAC.	N/A	User	AES keys (128, 192, 256 bits), HMAC keys (112 bits +)	R, W, EX
Module initialization	N/A	Starting of the host system and the module	N/A	Crypto officer	N/A	EX
Self-tests	HMAC-SHA-512	Integrity test of the kernel static binary performed by the sha512-hmac binary	Cisco Systems NSS Module	Crypto officer	HMAC-SHA512 key	R, EX

	AES HMAC SHA-1, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512 SHA-1, SHA-256, SHA-384, SHA-512	Algorithm self-tests tests	N/A			
Show status	N/A	Via verbose mode, exit codes and kernel logs	N/A	User	N/A	R, EX
Zeroize	N/A	Overwrite of in use CSPs.	N/A	Crypto officer	128, 192, and 256 bits AES keys, HMAC keys	R, EX
Installation and configuration	N/A	Installation on the host device by the Crypto officer	N/A	Crypto officer	N/A	R, EX

Table 4 – Available Cryptographic Module's Services in FIPS mode

The following table identifies the applicable CAVP certificates for each algorithm:

Algorithms	CAVS Cert(s).	Bounded Module
AES	C74	N/A
HMAC SHA-1, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512	C74	N/A
SHA-1, SHA-256, SHA-384, SHA-512	C74	N/A
HMAC SHA-512	C503	Cisco Systems NSS Module
SHA-512	C503	Cisco Systems NSS Module

Table 5 – CAVP Certificates

There are some algorithm modes/key sizes that were tested but not implemented by the module. Only the algorithms, modes, and key sizes that are implemented by the module are listed.

In non-Approved mode the Module supports the following non-FIPS Approved algorithms, which shall not be used in the FIPS Approved mode of operation:

Service	Algorithms	Note(s) / Mode(s)	Role	CSPs	Access
Symmetric encryption/ decryption	AES (non-compliant, not allowed and not approved for FIPS use)	XTS	User	128/256 bits AES keys	R, W, EX
Keyed hash	HMAC (non-compliant, not allowed and not approved for FIPS use)	Keys smaller than 112 bits	User	HMAC keys with size less than 112 bits	R, W, EX
Random number generation	ansi_cprng	N/A	User	seed	R, W, EX

Table 6 – Service Details for the non-FIPS mode

3.3. Authentication

The module is a Level 1 software-only cryptographic module and does not implement authentication. The role is implicitly assumed based on the service requested.

4. Physical Security

The module is comprised of software only and thus does not claim any physical security.

5. Operational Environment

5.1. Applicability

The module operates in a modifiable operational environment per FIPS 140-2 level 1 specifications. The module runs on a commercially available general-purpose operating system executing on the hardware specified in section 1.2.

5.2. Policy

The operating system is restricted to a single operator (concurrent operators are explicitly excluded). The application that request cryptographic services is the single user of the module, even when the application is serving multiple clients.

In FIPS approved mode, the ptrace(2) system call, the debugger (gdb(1)) and strace(1) shall not be used. In addition, other tracing mechanisms offered by the Linux environment, such as ftrace or kprobes (including systemtap) shall not be used.

6. Cryptographic Key Management

The application that uses the module is responsible for appropriate destruction and zeroization of the key material. The library provides functions for key allocation and destruction, which overwrites the memory that is occupied by the key information with “zeros” before it is deallocated.

6.1. CSPs/Keys

Here are listed the CSPs/keys details concerning storage, input, output, generation and zeroization:

Type	Keys/CSPs	Key Generation	Key Storage	Key Entry/Output	Key Zeroization
Symmetric keys	AES	N/A	Protected kernel memory	API allows caller on the same GPC to supply key	Memory is automatically overwritten by zeroes when freeing the cipher handler
HMAC keys	HMAC keys	N/A	Protected kernel memory	HMAC key can be supplied by calling application	Memory is automatically overwritten by zeroes when freeing the cipher handler
HMAC key used for integrity	HMAC key	N/A – Installed as part of the module	Persistently stored in plaintext as part of the sha512-hmac application	N/A - key is only used for integrity verification	Zeroized in memory by sha512hmac

Table 7 – Keys/CSPs

The module does not provide any key generation service or perform key generation for any of its Approved algorithms. Keys are passed in from calling application via API parameters.

6.2. Key / Critical Security Parameter (CSP) Access

An authorized application as user (the User role) has access to all key data during the operation of the module.

6.3. Key / CSP Storage

Symmetric keys are provided to the module by the calling process, and are destroyed when released by the appropriate API function calls. The module does not perform persistent storage of keys.

6.4. Key / CSP Zeroization

When a calling kernel component calls the appropriate API function that operation overwrites memory with 0s and then frees that memory.

7. EMI/EMC

The GPC(s) used during testing met Federal Communications Commission (FCC) FCC Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined by 47 Code of Federal Regulations, Part15, Subpart B. FIPS 140-2 validation compliance is maintained when the module is operated on other versions of the GPOS running in single user mode, assuming that the requirements outlined in NIST IG G.5 are met.

8. Self-Tests

FIPS 140-2 requires that the Module perform self-tests to ensure the integrity of the Module and the correctness of the cryptographic functionality at start up.

A failure of any of the self-tests panics the Module. The only recovery is to reboot. For persistent failures, you must reinstall the kernel. See section 9.1 for details.

No operator intervention is required during the running of the self-tests.

8.1. Power-Up Self-Tests

The module performs power-up self-tests at module initialization to ensure that the module is not corrupted and that the cryptographic algorithms work as expected. The self-tests are performed without any user intervention.

While the module is performing the power-up tests, services are not available, and input or output is not possible: the module is single-threaded and will not return to the calling application until the self-tests are completed successfully.

8.1.1. Integrity Tests

The Module performs power-up self-tests (at module initialization). Input, output, and cryptographic functions cannot be performed while the Module is in a self test or error state. The Module is single-threaded during the self tests and will stop the boot procedure, and therefore any subsequent operation before any other kernel component can request services from the Module.

The Crypto Officer with physical or logical access to the Module can run the POST (Power-On Self-Tests) on demand by power cycling the Module or by rebooting the operating system.

For Known Answer Test, HMAC SHA-512 provided by the NSS bound module is tested before the NSS module makes itself available to the sha512 HMAC application.

An HMAC SHA-512 (provided by the NSS bound module) calculation is performed on the module to verify integrity.

NOTE: The fact that the kernel integrity check passed, which requires the loading of sha512hmac with the self tests implies a successful execution of the integrity and self tests of sha512hmac (the HMAC is stored in /usr/lib/hmaccalc/sha512hmac.hmac).

With respect to the integrity check of kernel loadable components providing the cryptographic functionality, the fact that the self-test of these cryptographic components is displayed implies that the integrity checks of each kernel component passed successfully.

8.1.2. Known Answer Tests

The table below summarizes the power-on self tests performed by the module, which includes the Integrity Test of the module itself as stated above and the Known Answer Test for each approved cryptographic algorithm.

Algorithm	Test
AES	KAT, encryption and decryption are tested separately
HMAC SHA-1, -224, -256, -384, -512	KAT
SHA-1, -224, -256, -384, -512	KAT

Table 8 – Module Self-Tests

8.1.3. Conditional Tests

The module does not perform conditional test as no algorithm implemented require this testing.

9. Guidance

9.1. Cryptographic Officer Guidance

To operate the Kernel Crypto API module, the operating system must be restricted to a single operator mode of operation.

9.1.1. Secure Installation and Startup

Crypto Officers use the Installation instructions to install the Module in their environment.

To bring the Module into FIPS approved mode, perform the following:

- Install the dracut-fips package: install dracut-fips
- Recreate the INITRAMFS image: dracut -f

After regenerating the initramfs, the Crypto Officer has to append the following string to the kernel command line by changing the setting in the boot loader: fips=1

9.2. User Guidance

When using the Module, the user shall utilize the Linux Kernel Crypto API provided memory allocation mechanisms. In addition, the user shall not use the function copy_to_user() on any portion of the data structures used to communicate with the Linux Kernel Crypto API.

Only the cryptographic mechanisms provided with the Linux Kernel Crypto API are considered for use. The NSS bound module, although used, is only considered to support the integrity verification and is not intended for general-purpose use with respect to this Module.

9.3. Handling Self-Test Errors

A self-test failure within the Kernel Crypto API module will result in the module entering an error state panicking the kernel and the operating system will not load. Recover from this error state by trying to reboot the system. If the failure continues, you must reinstall the software package being sure to follow all instructions. The Kernel Crypto API module performs a power-on self test that includes an integrity check and known answer tests for the available cryptographic algorithms.

The kernel dumps self test success and failure messages into the kernel message ring buffer. Post boot, the messages are moved to /var/log/messages.

Use **dmesg** to read the contents of the kernel ring buffer. The format of the ringbuffer (**dmesg**) output is:

alg: self-tests for %s (%s) passed

10. Acronyms

AES	Advanced Encryption Standard
API	Application programming interface
CAVP	Cryptographic Algorithm Validation Program
CMVP	Cryptographic Module Validation Program
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standards
GPC	General Purpose Computer
GPOS	General Purpose Operating System
HMAC	keyed-hash message authentication code
POST	Power-On Self-Tests
SHA	Secure Hash Algorithm

Table 9 – Acronyms

--- End of Document ---