IOS Common Cryptographic Module (IC2M)

FIPS 140-2 Non Proprietary Security Policy
Level 1 Validation

Version 0.3

April 18, 2013

Table of Contents

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for Cisco's IOS Common Cryptographic Module (IC2M) Version Rel 1(1.0.0), Version Rel 1(1.0.1) and Version Rel 1(1.0.2). This security policy describes how the module meets the security requirements of FIPS 140-2 Level 1 and how to run the modules in a FIPS 140-2 mode of operation and may be freely distributed.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 — *Security Requirements for Cryptographic Modules*) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the NIST website at http://csrc.nist.gov/groups/STM/index.html.

1.2 Module Validation Level

The following table lists the level of validation for each area in the FIPS PUB 140-2.

No.	Area Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	1
6	Operational Environment	N/A
7	Cryptographic Key management	1
8	Electromagnetic Interface/Electromagnetic Compatibility	1
9	Self-Tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A
	Overall module validation level	**1**

Table 1 Module Validation Level

1.3 References

This document deals only with operations and capabilities of the IC2M listed above in section 1 in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the routers from the following sources:

For answers to technical or sales related questions please refer to the contacts listed on the Cisco Systems website at www.cisco.com.

The NIST Validated Modules website (http://csrc.nist.gov/groups/STM/cmvp/validation.html) contains contact information for answers to technical or sales-related questions for the module.

1.4 Terminology

In this document, Cisco IOS Common Cryptographic Module is referred to as IC2M or the module.

1.5 Document Organization

The Security Policy document is part of the FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

Vendor Evidence document
Finite State Machine
Other supporting documentation as additional references

This document provides an overview of the IC2M identified in section 1 above and explains the secure configuration and operation of the module. This introduction section is followed by Section 2, which details the general features and functionality of the appliances. Section 3 specifically addresses the required configuration for the FIPS-mode of operation.

With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Submission Documentation is Cisco-proprietary and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Cisco Systems.

2 Cisco IOS Common Cryptographic Module (IC2M)

This module provides the FIPS validated cryptographic algorithms for services requiring those algorithms. The module does not implement any protocols directly. Instead, it provides the cryptographic primitives and functions to allow IOS to implement those various protocols.

2.1 Cryptographic Module Characteristics

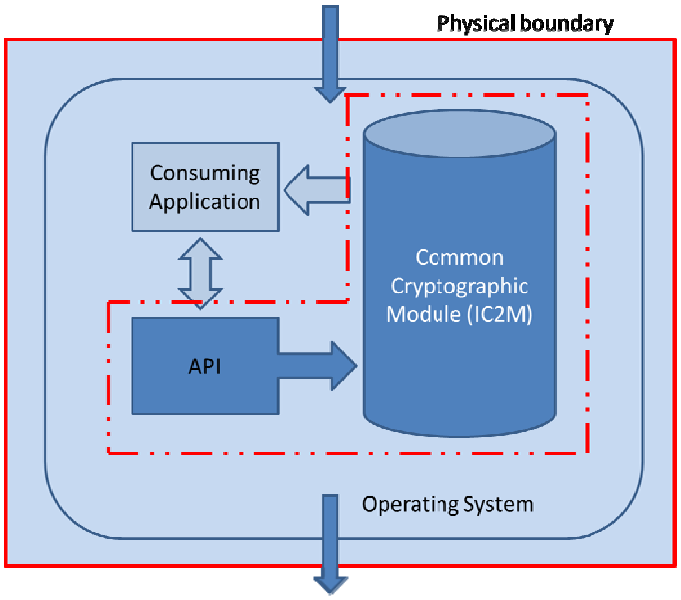

Figure 1 - Cisco IC2M logical block diagram

The module's logical block diagram is shown in Figure 1 above. The red dashed line area denotes the logical cryptographic boundary of the module. While the red solid line denotes the physical cryptographic boundary of the module which is the enclosure of the system on which the module is executed.

The IC2M is a single binary object file, sub_crypto_ic2m_k9.o (Cisco IOS) classed as a multi-chip standalone firmware, cryptographic module. It is capable of being utilized and is validated on any platform running Cisco IOS containing Power-PC 405 250MHz, Intel Woodcrest 2.13GHz, and PMC RM5261A MIPS processor 350MHz architecture. This module was tested on the following Cisco hardware platforms:

- Cisco Catalyst 2960 with IOS 15.0 - Power-PC 405 250MHz

- Cisco 3925 ISR with IOS 15.2 - Intel Woodcrest 2.13GHz

- Cisco 2811 ISR with IOS 15.2 - PMC RM5261A MIPS

2.2 Module Interfaces

The physical ports of the Module are the same as the system on which it is executing. The logical interface is a C-language application program interface (API).

The Data Input interface consists of the input parameters of the API functions. The Data Output interface consists of the output parameters of the API functions. The Control Input interface consists of the actual API functions. The Status Output interface includes the return values of the API functions.

The module provides logical interfaces to the system, and is mapped to the following FIPS 140-2 defined logical interfaces: data input, data output, control input, status output, and power. The logical interfaces and their mapping are described in the following table:

Interface	Description
Date Input	API input parameters plaintext and/or ciphertext data
Data Output	API output parameters plaintext and/or ciphertext data
Control Input	API function calls function calls, or input arguments that specify commands and control data used to control the operation of the module
Status Output	API return codes

	function return codes, error codes, or output arguments that receive status information used to indicate the status of the module

Table 1 - Logical Interfaces Details

2.3 Roles and Services

The Module meets all FIPS 140-2 level 1 requirements for Roles and Services, implementing both Crypto Officer and User roles, which are classed as processes. As allowed by FIPS 140-2, the Module does not support user authentication for these roles, which is handled by the system implementing IC2M. Only one role may be active at a time and the Module does not allow concurrent operators.

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

- Installation of the Crypto Module which is embedded in the IOS image and installed on the IOS platform is assumed implicitly as the Crypto Officer when install occurs.

The services available only in FIPS mode to the Crypto Officer and User roles consist of the following:

Services	Access	CSPs	Crypto Officer	User
Encryption/decryption	execute	Symmetric keys AES, Triple-DES	**X**	**X**
Hash (HMAC)	execute	HMAC SHA-1 key, HMAC-SHA-1	**X**	**X**
Key agreement	execute	DH and ECDH private key	**X**	**X**
Key generation	Write/execute	Symmetric key AES, Triple-DES	**X**	**X**
Key transport	execute	Asymmetric private key RSA	**X**	**X**
Message Digest (SHS)	execute	None	**X**	**X**
Perform Self-Tests	Execute/read	N/A	**X**	**X**
Random number generator	execute	Seed key, seed	**X**	**X**
Show Status	Execute	N/A	**X**	**X**
Signature	execute	Asymmetric private key ECDSA, RSA	**X**	**X**
Zeroization	execute	Symmetric key, asymmetric key, HMAC-SHA-1 key, seed key, seed	**X**	**X**

Table 2 – Services

2.4 Physical Security

The module obtains its physical security from any platform running Cisco IOS with production grade components as allowed by FIPS 140-2 level 1.

2.5 Cryptographic Key Management

Keys that reside in internally allocated data structures can only be accessed using the Module defined API. The operating system protects memory and process space from unauthorized access. Zeroization of sensitive data is performed automatically by API function calls for intermediate data items, and on demand by the calling process using the module provided API function calls provided for that purpose.

Zeroization consists of overwriting the memory that store the key or refreshing the volatile memory. Keys can also be zeroized by cycling the power.

The module supports the following keys and critical security parameters (CSPs):

Key/CSP Name	Generation/ Algorithm	Description	Storage and Zeroization
Asymmetric Key	RSA, ECDSA	RSA: 1024-4096 bits ECDSA: P-256, P-384 Used for signature generation/verification RSA: Also used for key transport	Stored and zeroized outside the module in the host OS
Diffie-Hellman private exponent	DH	DH: 1024-4096 bits Used for key agreement	Stored and zeroized outside the module in the host OS
DRBG Key	DRBG SP 800-90	DRBG key for DRBG	Zeroized with generation of new seed
DRBG Seed	DRBG SP 800-90	Seed for DRBG	Zeroized with generation of new seed
DRBG V Value	DRBG SP 800-90	Random value generated internally	Zeroized with generation of new value
EC Diffie-Hellman private exponent	ECDH	ECDH: P-192, P-256, P-384, P-521 Used for key agreement	Stored and zeroized outside the module in the host OS
Firmware integrity key	HMAC	integrity test at power-on key Embedded within module	Stored in plaintext and zeroized by uninstalling the module

HMAC-SHS Key	FIPS 198	SHA-1, 256, 384 and 512 Message authentication code key	Stored and zeroized outside the module in the host OS
Symmetric Key	AES, Triple-DES	AES: 128, 192, 256 bits Triple-DES: 168 bits Used for symmetric encryption/decryption	Stored and zeroized outside the module in the host OS

Table 3 - Cryptographic Keys and CSPs

All cryptographic 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.

2.6 Cryptographic Algorithms

The module implements a variety of approved and non-approved algorithms.

Approved Cryptographic Algorithms

The cryptographic module supports the following FIPS-140-2 approved algorithm implementations:

Algorithm	Algorithm Certificate Number
AES	2134 and 2136
DRBG	237
ECC KAS	30
ECDSA	322
FFC KAS	30
HMAC SHS	1304
RSA	1100
SHS	1858 and 1859
Triple-DES	1358, 1359 and 1360

Table 4 - Approved Cryptographic Algorithms

Non-FIPS Approved Algorithms Allowed in FIPS Mode:

The module supports the following non-FIPS approved algorithms which are permitted for use in the FIPS approved mode:

- Diffie-Hellman (CVL Cert. #30, key agreement; key establishment methodology provides between 80 and 112 bits of encryption strength)

- EC Diffie-Hellman (CVL Cert. #30, key agreement; key establishment methodology provides between 80 and 256 bits of encryption strength)
- RSA (key wrapping; key establishment methodology provides between 80 and 112 bits of encryption strength)

Non-Approved Cryptographic Algorithms

- DES
- HMAC-MD5
- MD2
- MD5
- RC2
- RC4
- SEAL

2.7 Self-Tests

The modules include an array of self-tests that are run automatically during startup and periodically when called during operations to prevent any secure data from being released and to insure all components are functioning correctly.

Self-tests performed

- IC2M Self Tests
    - POSTs - IOS Common Crypto Module
        - Firmware Integrity Test (HMAC SHA-256)
        - AES KAT
        - AES GCM KAT
        - AES-CMAC KAT
        - DRBG KAT
        - ECDSA Sign/Verify
        - HMAC-SHA-1 KAT
        - HMAC-SHA-256 KAT
        - HMAC-SHA-384 KAT
        - HMAC-SHA-512 KAT
        - KAS ECC Primitive “Z” KAT
        - KAS FFC Primitive “Z” KAT
        - RSA KAT
        - SHA-1 KAT
        - SHA-256 KAT
        - SHA-384 KAT
        - SHA-512 KAT

        - Triple-DES KAT
    - POSTs - IOS Common Crypto Module-Extended
        - AES KAT
        - SHA-1 KAT
        - SHA-256 KAT
        - SHA-384 KAT
        - SHA-512 KAT
        - Triple-DES KAT
    - POSTs - IOS Common Crypto Module-Extended2
        - Triple-DES KAT
- Conditional tests
        - Pairwise consistency test for RSA
        - Pairwise consistency test for ECDSA
        - Continuous random number generation test for approved DRBG and non-approved RNG

The module inhibits all access to cryptographic algorithms during initialization and self-tests due to the process architecture in use. Additionally, the power-on self-tests are performed after the cryptographic systems are initialized but prior to the underlying OS initialization of external interfaces; this prevents the security appliances from passing any data before completing self-tests and entering FIPS mode. In the event of a power-on self-test failure, the cryptographic module will force the IOS platform to reload and reinitialize the operating system and cryptographic module. This operation ensures no cryptographic algorithms can be accessed unless all power on self-tests are successful.

In addition to the automatic operation at cryptographic module initialization time, self-tests can also be initiated on demand by the Crypto Officer or User by issuing the operating system command which invokes the module's "crypto_engine_nist_run_self_tests() function."

3 Secure Operation of the IC2M

The module is completely and permanently embedded into the greater IOS operating system. There are no installation considerations besides the typical loading of the larger IOS system. That is, the imbedded IC2M firmware module cannot be modified, replaced or upgraded except by loading a new IOS version in its entirety.

The Module functions entirely within the process space of the process that invokes it, and thus satisfies the FIPS 140-2 requirement for a single user mode of operation.

The following policy must always be followed in order to achieve a FIPS 140-2 mode of operation:

• Calling the function ic2m_init() initializes the cryptographic module and runs all required power-on self-tests, placing the module in the FIPS-approved mode of operation.

• Only FIPS approved or allowed algorithms and key sizes may be used. Please refer to section 2.6 for more information.

Upon initialization of the Module, the module will run its power-up self-tests. Successful completion of the power-up self-tests indicates the module has passed the self-tests and is ready within the IOS. If an error occurs during the self-test the module outputs the following message:

requesting a reload of the OS. %CRYPTO-0-SELF_TEST_FAILURE: Encryption self-test failed **(**<*failing test description*>**)** where <*failing test description*> identifies the name of the self-test that failed.