# ISO/IEC 19790 and FIPS 140-3

# Non-Proprietary Security Policy
# for
# IOS Common Cryptographic Module (IC2M)
# Firmware Version: Rel5b

Last Updated: August 6, 2024
Version 1.4

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

# Table of Contents

# List of Figures

# List of Tables

# 1 General

This document is Cisco's non-proprietary security policy for the IOS Common Cryptographic Module (IC2M) with firmware version Rel5b (herein referred to as "IC2Mrel5b" or the "module"). The following details how this module meets the security requirements of FIPS[1] 140-3, NIST[2] SP[3] 800-140, and ISO[4]/IEC[5] 19790 for a Security Level 1 Firmware cryptographic module.

The security requirements cover areas related to the design and implementation of a cryptographic module. Table 1 below indicates the security level for each area of the module.

**Table 1 - Security Levels**

| ISO/IEC 24759:2017 Section 6 | FIPS 140-3 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 |

The overall security level of the module is 1.

[1] FIPS – Federal Information Processing Standards
[2] NIST – National Institute of Standards and Technology
[3] SP – Special Publication
[4] ISO – International Organization of Standardization
[5] IEC – International Electrotechnical Commission

# 2 Cryptographic module specification

IC2Mrel5b is a single binary object file (sub_crypto_ic2m_k9.o) and is classified as a multi-chip standalone firmware module.

IC2Mrel5b is a cryptographic library that supports cryptographic operations executed by a calling application. The calling application leverages the module's well-defined API[6] to initialize the module and call cryptographic algorithms for encryption/decryption, key generation, signature generation/verification, and hashing. The cryptographic library does not implement any protocols, but does provide the cryptographic primitives for IPsec[7]/IKE[8]v2, SNMP[9]v3, SRTP[10], SSH[11]v2, and TLS[12] v1.2/v1.3. No SSP[13]s are stored within the cryptographic boundary of the module.

The module is intended for use on any Cisco device that runs the IOS-XE OS[14], so the physical perimeter of the module is the testing platform. The module's operational environment is non-modifiable.

Table 2 below lists the tested operational environments.

**Table 2 – Tested Operational Environments**

| # | Operating Systems | Hardware Platform | Processor (Acceleration) | PAA/Acceleration |
|---|---|---|---|---|
| 1 | IOS-XE 17.12 | Cisco Aggregated Services Router (ASR) 1001-HX | Intel Xeon E3-1125C v2 | N/A[15] |

Table 3 below lists vendor affirmed operational environments.

**Table 3 - Vendor Affirmed Operational Environments**

| # | Operating System | Hardware Platform |
|---|---|---|
| 1 | IOS-XE 17.12 | Catalyst 9200 Series Switches |
| 2 | IOS-XE 17.12 | Catalyst 9300 Series Switches |
| 3 | IOS-XE 17.12 | Catalyst 9400 Series Switches |
| 4 | IOS-XE 17.12 | Catalyst 9500 Series Switches |
| 5 | IOS-XE 17.12 | Catalyst 9600 Series Switches |
| 6 | IOS-XE 17.12 | Cisco Embedded Services 3300 Series Switch |
| 7 | IOS-XE 17.12 | Cisco Embedded Services 9300 Series Switch |
| 8 | IOS-XE 17.12 | Cisco Catalyst Industrial Ethernet 3000 Series Switch |
| 9 | IOS-XE 17.12 | Cisco Catalyst Industrial Ethernet 9300 Series Switch |
| 10 | IOS-XE 17.12 | Cisco C8500, C8500L Series Edge Platforms |
| 11 | IOS-XE 17.12 | Cisco C8200, C8200L, C8300 Series Edge Platforms |
| 12 | IOS-XE 17.12 | Cisco Aggregation Services Router (ASR) 1000 series |
| 13 | IOS-XE 17.12 | Cisco Integrated Services Router (ISR) 4000 series |
| 14 | IOS-XE 17.12 | Cisco Integrated Services Router (ISR) 1000 series |
| 15 | IOS-XE 17.12 | Cisco C8000V Edge Software Router |
| 16 | IOS-XE 17.12 | Cisco IR 1100, 1800, 8100, 8300 Series Industrial Routers |

[6] APIs – Application Programming Interface
[7] IPsec – IP Security
[8] IKE – Internet Key Exchange
[9] SNMP – Simple Network Management Protocol
[10] SRTP – Secure Real-time Transport Protocol
[11] SSH – Secure Shell
[12] TLS – Transport Layer Security
[13] SSP – Sensitive Security Parameters
[14] OS – Operating System
[15] N/A – Not Applicable

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

## Modes of operation

The module supports both approved and non-approved mode of operation. The module will be in approved mode when all pre-operational self-tests have completed successfully and only approved algorithms/services are invoked. Table 4 and Table 7 below for a list of the supported approved/allowed algorithms/services. The non-approved mode is entered when a non-approved algorithm/non-approved service is invoked. See Table 5 and Table 9 below for a list of non-approved algorithms/non-approved services. The Approved mode of operation can only be transitioned into the non-Approved mode by calling one of the non-Approved services listed in Table 9 - Non-Approved Services.

The following tables list all Approved or Vendor-affirmed security functions of the module, including specific key size(s) -in bits otherwise noted- employed for approved services, and implemented modes of operation.

**Table 4 - Approved Algorithms**

| CAVP[16] Cert | Algorithm/Standard | Mode/Method | Description/Key Size(s)/Key strength(s) | Use/Function |
|---|---|---|---|---|
| A4354 | AES[17]<br>[FIPS PUB[18] 197]<br>[NIST SP 800-38A] | AES-CBC[19]; | Key Length: 128, 192, 256 bits | Symmetric encryption and decryption |
| A4354 | AES<br>[FIPS PUB 197]<br>[NIST SP 800-38A] | AES-CFB[20]128; | Key Length: 128, 192, 256 bits | Symmetric encryption and decryption |
| A4354 | AES<br>[FIPS PUB 197]<br>[NIST SP 800-38A] | AES-ECB[21] | Key Length: 128, 192, 256 bits | Symmetric encryption and decryption |
| A4354 | AES<br>[FIPS PUB 197]<br>[NIST SP 800-38B] | AES-CMAC[22] | Key Length: 128, 256 bits | Authenticated symmetric encryption and decryption |
| A4354 | AES<br>[FIPS PUB 197]<br>[NIST SP 800-38D] | AES-GCM[23] | Key Length: 128, 192, 256 bits | Authenticated symmetric encryption and decryption |
| A4354 | AES<br>[FIPS PUB 197]<br>[NIST SP 800-38D] | AES-GMAC[24] | Key Length: 128 bits | Authenticated symmetric encryption and decryption |
| A4354 | AES<br>[NIST SP 800-38F] | AES-KW[25]<br>(encrypt/decrypt) | Key Length: 128, 192, 256 bits | Symmetric encryption and decryption |
| A4354 | ECDSA<br>[FIPS PUB 186-4] | ECDSA KeyGen | Curves: P-256, P-384, P-521 | ECDSA keypair generation |

[16] CAVP – Cryptographic Algorithm Validation Program
[17] AES – Advanced Encryption Standard
[18] PUB - Publication
[19] CBC – Cipher-Block Chaining
[20] CFB – Cipher Feedback
[21] ECB – Electronic Codebook
[22] CMAC – Cipher-based Message Authentication Code
[23] GCM – Galois Counter Mode
[24] GMAC – Galois Message Authentication Code
[25] KW – Key Wrap

| CAVP[16] Cert | Algorithm/Standard | Mode/Method | Description/Key Size(s)/Key strength(s) | Use/Function |
|---|---|---|---|---|
| A4354 | ECDSA<br>[FIPS PUB 186-4] | ECDSA KeyVer | Curves: P-256, P-384, P-521 | ECDSA keypair verification |
| A4354 | ECDSA<br>[FIPS PUB 186-4] | ECDSA SigGen | Curves: P-256, P-384, P-521 | ECDSA signature generation |
| A4354 | ECDSA<br>[FIPS PUB 186-4] | ECDSA SigVer | Curves: P-256, P-384, P-521 | ECDSA signature verification |
| A4354 | RSA[26]<br>[FIPS PUB 186-4] | RSA KeyGen:<br>- Mode: B.3.4<br>- 2048/3072 bits with SHA2-256 | Modulus: 2048, 3072, 4096 | RSA keypair generation |
| A4354 | RSA<br>[FIPS PUB 186-4] | RSA SigGen:<br>- PKCSv1.5<br>- 2048/3072 bits with SHA2-256/384/512 | Modulus:<br>2048, 3072, 4096 | RSA signature generation |
| A4354 | RSA<br>[FIPS PUB 186-4] | RSA SigVer:<br>- PKCSv1.5<br>- 2048/3072 bits with SHA2-256/384/512 | Modulus:<br>2048, 3072, 4096 | RSA signature verification |
| A4354 | KAS[27]-ECC-SSC[28]<br><br>[NIST SP 800-56Arev3] | KAS-ECC29-SSC<br><br>Scheme:<br>dhEphem:<br>KAS Role: initiator, responder | Curves: P-256, P-384, P-521 | Key establishment methodology provides between 128 and 256 bits of encryption strength |
| A4354 | KAS-FFC-SSC<br>[NIST SP 800-56Arev3] | KAS-FFC[30]-SSC<br><br>Scheme:<br>dhEphem:<br>KAS Role: initiator, responder | MODP-2048, MODP-3072, MODP-4096 | Key establishment methodology provides between 112 and 152 bits of encryption strength |
| A4354 | Safe Primes Key Generation<br>[NIST SP 800-56Arev3] | KeyGen for DH[31] (CKG using method in Sections 4 and 5.1 of SP 800-133rev2) | MODP-2048, MODP-3072, MODP-4096 | KAS-FFC Keypair domain parameters generation |
| A4354 | KDF IKEv2<br>[NIST SP800-135rev1] (CVL) | KDF IKEv2 | N/A | Key derivation function used in IKEv2 |
| A4354 | KDF SNMP<br>[NIST SP800-135rev1] (CVL) | KDF SNMP | N/A | Key derivation function used in SNMPv3 |
| A4354 | KDF SRTP<br>[NIST SP800-135rev1] (CVL) | KDF SRTP | N/A | Key derivation function used in SRTP |
| A4354 | KDF SSH<br>[NIST SP800-135rev1] (CVL) | KDF SSH | N/A | Key derivation function used in SSHv2 |

[26] RSA – Rivest, Shamir, and Adleman
[27] KAS – Key Agreement Scheme
[28] SSC – Shared Secret Computation
[29] ECC – Elliptic Curve Cryptography
[30] FFC – Finite Field Cryptography
[31] DH – Diffie-Hellman

| CAVP[16] Cert | Algorithm/Standard | Mode/Method | Description/Key Size(s)/Key strength(s) | Use/Function |
|---|---|---|---|---|
| A4354 | TLSv1.2 KDF RFC7627 (CVL) | TLSv1.2 KDF RFC7627 | N/A | Key derivation in TLSv1.2 with RFC 7627 KDF with Extended Master Secret |
| A4354 | TLS v1.3 KDF<br>[RFC 8446] (CVL) | TLSv1.3 KDF | N/A | Key derivation function used in TLSv1.3 |
| A4354 | HMAC<br>[FIPS 198-1] | HMAC-SHA-1 | Key Length: 112-bits or greater | Keyed hash |
| A4354 | HMAC<br>[FIPS 198-1] | HMAC-SHA2-256 | Key Length: 112-bits or greater | Keyed hash |
| A4354 | HMAC<br>[FIPS 198-1] | HMAC-SHA2-384 | Key Length: 112-bits or greater | Keyed hash |
| A4354 | HMAC<br>[FIPS 198-1] | HMAC-SHA2-512 | Key Length: 112-bits or greater | Keyed hash |
| A4354 | SHS<br>[FIPS PUB 180-4] | SHA-1 | N/A | Message digest<br><br>Note: SHA-1 is not used for digital signature generation |
| A4354 | SHS<br>[FIPS PUB 180-4] | SHA2-256 | N/A | Message digest |
| A4354 | SHS<br>[FIPS PUB 180-4] | SHA2-384 | N/A | Message digest |
| A4354 | SHS<br>[FIPS PUB 180-4] | SHA2-512 | N/A | Message digest |
| A4354 | DRBG[32]<br>[NIST SP 800-90Arev1] | CTR[33]_DRBG (AES-256)<br>Derivation Function Enabled: Yes | 256 bits | Random number generation |
| Vendor Affirmed | CKG<br>[SP 800-133rev2] | N/A | N/A | Symmetric and asymmetric key generation (Please refer to section "SSP Generation" in this document for more information) |

NOTES:

- There are algorithms, modes, and key moduli sizes that have been CAVP-tested but are not used by any approved services of the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in the tables above are used by an approved service of the module.
- The module supports generation of ECDSA, RSA, ECDH[34], and DH asymmetric key pairs in accordance with NIST SP 800-133r2, section 5. The module supports generation of AES and HMAC symmetric keys in accordance with NIST SP 800-133r2, section 6.

[32] DRBG – Deterministic Random Bit Generator

[33] CTR - Counter

- No parts of IPsec/IKEv2, SNMPv3, SRTP SSH, and TLS protocols, other than the KDFs, have been tested by the CAVP and CMVP.

Table 5 below lists all non-Approved algorithms not allowed in the approved mode of operation implemented by the module.

**Table 5 – Non-Approved Algorithms Not Allowed in the Approved Mode of Operation**

| Algorithm/Function | Use/Function |
|---|---|
| RSA | Key establishment with PKCS1-v1.5 padding |
| MD5 | Message digest |
| Triple-DES | Encryption/decryption |

In addition, the module does not implement Non-Approved Algorithms Allowed in Approved Mode of Operation and Non-Approved Algorithms Allowed in Approved Mode of Operation with No Security Claimed.

### Cryptographic Boundary

Figure 1 below depicts the cryptographic boundary (red dashed line) and physical perimeter (solid red line). The cryptographic boundary is defined as the cryptographic library. The physical perimeter is the Tested Operational Environment's Physical Perimeter (TOEPP) on which the module runs.

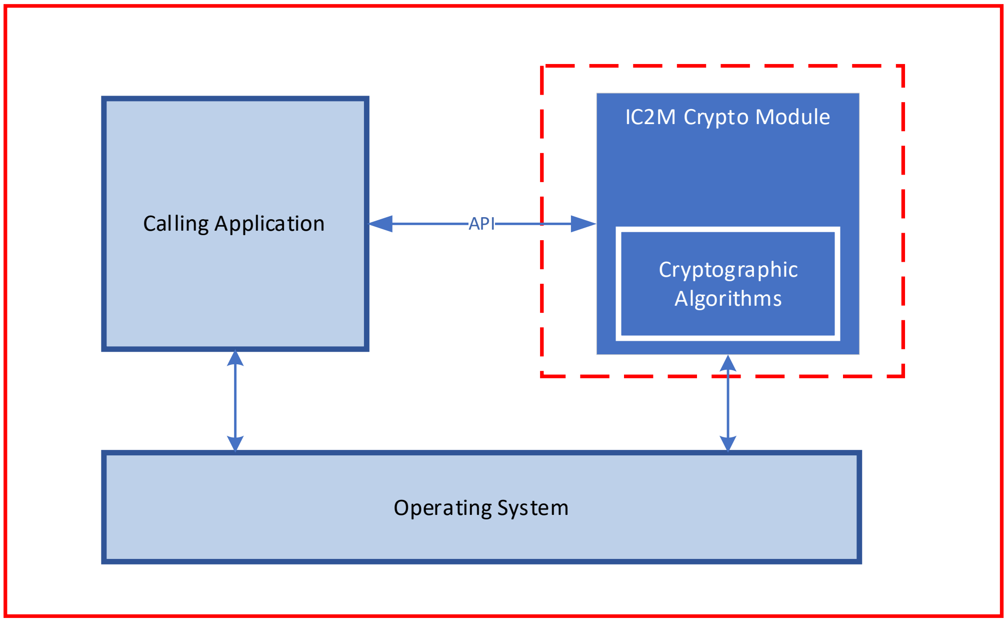

**Figure 1 – Module's boundary**

## 3 Cryptographic module interfaces

The module's physical perimeter encompasses the case of the tested platform mentioned in section 2 above. No data passes in or out of these physical ports. The module provides logical interfaces via well-defined APIs. The logical interfaces provided by the module are mapped to the following FIPS 140-3 interfaces:

- Data input
- Data output
- Control input
- Control output
- Status output

Table 6 below provides a description of data that passes through each logical interface.

**Table 6 – Ports and Interfaces**

| Physical Port | Logical Interface | Data that passes over port/interface |
|---|---|---|
| N/A | Data Input | API input parameters – plaintext and/or ciphertext data |
| N/A | Data Output | API output parameters – plaintext and/or ciphertext data |
| N/A | Control Input | API input parameters – function calls, or input arguments that specify commands and control data used to control the operation of the module |
| N/A | Control Output | Not applicable |
| N/A | 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 |
| N/A | Power | Not applicable |

## 4 Roles, services, and authentication

The module supports the CO[35] role. The module does not provide any authentication methods. The module does not allow concurrent operators. The CO role is implicitly assumed based on the service requested. The module provides the services in Table 8 below to the CO.

Table 7 below provides roles, service commands, input, and output for the module.

**Table 7 – Roles, Service Commands, Input and Output**

| Role | Service | Input | Output |
|---|---|---|---|
| CO | Initialize module | None | Module loaded |
| CO | Show status | API command | Module's current status |
| CO | Perform self-test | API command | Output display on each algorithm running self-test and pass/fail indicator |
| CO | Show version information | API command | Displays the module's name/ID and versioning information |
| CO | Symmetric cipher operation | API commands, key, and plaintext/ciphertext data | Plaintext or ciphertext |
| CO | Asymmetric cipher operation | API commands, keys, and ciphertext/plaintext data | Signature and plaintext/ciphertext |
| CO | Key exchange/agreement component (NIST SP 800-56Arev3) | API commands, asymmetric keys | Asymmetric key or key agreement component |
| CO | Key wrapping (KW) | API commands, wrapping key, key | Wrapped key |
| CO | Key derivation function | IKEv2 (existing application specific): IKEv2 parameters | KDF shared secret data |
| | | SNMPv3 (existing application specific): SNMPv3 parameters | KDF shared secret data |
| | | SRTP (existing application specific): SRTP parameters | KDF shared secret data |

[35] CO – Crypto Officer

| Role | Service | Input | Output |
|---|---|---|---|
| | | SSH (existing application specific): SSH parameters | KDF shared secret data |
| | | TLS (KDF, existing application specific): TLS parameters | KDF shared secret data |
| CO | Keyed hashing function | API commands, HMAC key, plaintext | MAC value |
| CO | Message digest | API commands, plaintext | Hash value |
| CO | Random number generation | API commands | Random bits |
| CO | Zeroization | API command | None |

Table 8 below lists all approved services that can be used in the approved mode of operation. The abbreviations of the access rights to keys and SSPs have the following interpretation:

- **G = Generate**: The module generates or derives the SSP.
- **R = Read**: The SSP is read from the module.
- **W = Write**: The SSP is updated, imported, or written to the module.
- **E = Execute**: The module uses the SSP in performing a cryptographic operation.
- **Z = Zeroize**: The module zeroizes the SSP.
- **N/A** = The service does not access any SSP during its operation.

**Table 8 - Approved Services**

| Service | Description | Approved Security Function | Keys and/or SSPs | Roles | Access Rights | Indicator |
|---|---|---|---|---|---|---|
| Initialize module | Initialization occurs when the module is loaded | None | None | CO | R | N/A |
| Show status | Display running status of the module | N/A | None | CO | N/A | N/A |
| Perform self-test | Perform Self-Tests (Pre-operational self-tests and Conditional Self-Tests) | AES-CBC;<br>AES-CMAC;<br>AES-GCM;<br>CTR-DRBG;<br>ECDSA SigGen;<br>ECDSA SigVer;<br>HMAC-SHA2-256;<br>KAS-FFC-SSC;<br>KAS-ECC-SSC;<br>RSA SigGen;<br>RSA SigVer;<br>SHA-1;<br>SHA2-256;<br>SHA2-512;<br>KDF IKEv2;<br>KDF SNMP;<br>KDF SRTP;<br>KDF SSH;<br>TLS v1.2 KDF with RFC7627;<br>TLS v1.3 KDF | Firmware integrity key (non-SSP) | CO | R | API return value |
| Show version information | Provide module’s name and version information | N/A | None | CO | N/A | N/A |
| Symmetric cipher operation | Perform encryption/decryption of data | CKG;<br>DRBG;<br>AES-CBC;<br>AES-CFB128;<br>AES-ECB; | DRBG entropy input;<br>DRBG seed; | CO | G, R, W, E | API return value |

| Service | Description | Approved Security Function | Keys and/or SSPs | Roles | Access Rights | Indicator |
|---|---|---|---|---|---|---|
| | | AES-CMAC;<br>AES-GMAC;<br>AES-GCM | DRBG internal state V value;<br>DRBG key;<br>AES EDK | | | |
| Asymmetric cipher operation | Perform signature generation/verification and key generation | CKG;<br>DRBG;<br>RSA KeyGen;<br>RSA SigGen;<br>RSA SigVer;<br>ECDSA KeyGen;<br>ECDSA KeyVer;<br>ECDSA SigGen;<br>ECDSA SigVer | DRBG entropy input;<br>DRBG seed;<br>DRBG internal state V value;<br>DRBG key;<br>RSA SGK;<br>RSA SVK;<br>ECDSA SGK;<br>ECDSA SVK | CO | G, R, W, E | API return value |
| Key exchange/agreement component | Perform key agreement primitives on behalf of the calling application (does not establish keys into the module) | CKG;<br>DRBG;<br>KAS-ECC-SSC;<br>KAS-FFC-SSC;<br>Safe Primes Key Generation | DRBG entropy input;<br>DRBG seed;<br>DRBG internal state V value;<br>DRBG key;<br>DH public key;<br>DH private key;<br>ECDH public key;<br>ECDH private key | CO | G, R, W, E | API return value |
| Key wrapping (KW) | Encrypt a key value on behalf of the calling application | CKG;<br>DRBG;<br>AES-KW | DRBG entropy input;<br>DRBG seed;<br>DRBG internal state V value;<br>DRBG key;<br>AES KWK | CO | G, R, W, E | API return value |
| KDF IKEv2 function | Derive keys for IKEv2 protocol | KDF IKEv2 | IKEv2 KDF Secret | CO | G, R, W, E | API return value |
| KDF SNMPv3 function | Derive keys for SNMPv3 protocol | KDF SNMP | SNMP KDF secret | CO | G, R, W, E | API return value |
| KDF SRTP function | Derive keys for SRTP protocol | KDF SRTP | SRTP KDF secret | CO | G, R, W, E | API return value |

| Service | Description | Approved Security Function | Keys and/or SSPs | Roles | Access Rights | Indicator |
|---|---|---|---|---|---|---|
| KDF SSHv2 function | Derive keys for SSHv2 protocol | KDF SSH | SSH KDF secret | CO | G, R, W, E | API return value |
| KDF TLS v1.2 function | Derive keys for TLS v1.2 protocol | TLSv1.2 KDF RFC 7627 | TLS v1.2 KDF extended master secret | CO | G, R, W, E | API return value |
| KDF TLS v1.3 function | Derive keys for TLS v1.3 protocol | TLSv1.3 KDF | TLS v1.3 KDF secret | CO | G, R, W, E | API return value |
| Keyed hashing function | Generate keyed hash | CKG;<br>DRBG;<br>HMAC-SHA-1;<br>HMAC-SHA2-256;<br>HMAC-SHA2-384;<br>HMAC-SHA2-512; | DRBG entropy input;<br>DRBG seed;<br>DRBG internal state V value;<br>DRBG key;<br>HMAC key | CO | G, R, W, E | API return value |
| Message digest | Generate message digest (secure hashing function) | SHA-1;<br>SHA2-256;<br>SHA2-384;<br>SHA2-512; | None | CO | G, R, W, E | API return value |
| Random number generation | Provide random data for key generation | CTR-DRBG | DRBG entropy input;<br>DRBG seed;<br>DRBG internal state V value;<br>DRBG key | CO | G, R, W, E | API return value |
| Zeroization | Zeroize all SSPs stored in allocated memory. Cleanup is the responsibility of the calling application. | None | All SSPs | CO | Z | N/A |

Table 9 below lists non-Approved services supported by the module.

**Table 9 - Non-Approved Services**

| Service | Description | Algorithm Accessed | Role | Indicator |
|---|---|---|---|---|
| Message digest | Generate message digest | MD5 | CO | API return value |
| Asymmetric cipher operation | Perform signature generation/verification | RSA (PKCS1-v1.5 padding) | CO | API return value |
| Symmetric cipher operation | Perform decryption of ciphertext data | Triple-DES | CO | API return value |

# 5 Software/Firmware security

### Integrity techniques

The IC2Mrel5b cryptographic module is a binary file (sub_crypto_ic2m_k9.o) statically linked within the IOS-XE OS. To ensure firmware security, the module is protected by an HMAC-SHA2-256 (HMAC Cert. #A4354) algorithm. The firmware integrity test key (non-SSP) was preloaded to the module's binary at the factory and used only for the pre-operational firmware integrity self-test. During initialization of the module, the integrity of the runtime executable is verified using an HMAC-SHA2-256 which is compared to a value computed at build time. If at load time the MAC does not match the stored, known MAC value, the module enters a critical error state where all crypto functionality inhibited. The module must be reloaded to attempt the integrity test again.

### Integrity test 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 power-cycle or reboot the tested platform to initiate the integrity test on-demand.

# 6 Operational environment

The module is operated in a non-modifiable operational environment per ISO/IEC 19790, section 7.6, level 1 specifications. The module is delivered as part of the IOS-XE OS. The OS is restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded). The application that makes calls to the module is the single user of the module. The module's firmware version running on each tested platform is Rel5b.

# 7 Physical security

Per ISO/IEC 19790, section 7.7, the module is defined as a multi-chip standalone firmware cryptographic module. The module runs on a host appliance made of production-grade components with standard passivation techniques.

# 8 Non-invasive security

At the time of publication of this Security Policy, non-invasive security is not required for FIPS 140-3 certification (see NIST SP 800-140F). The requirements of this area are not applicable to the module.

# 9 Sensitive security parameters management

Table 10 below provides information for the SSPs that are used by the cryptographic services implemented in the module.

**Table 10 - SSPs**

| Keys/ SSP Name/Type | Strength | Security Function and Cert Number | Generation | Import/ Export | Establish-ment | Storage | Zeroization | Use and released keys |
|---|---|---|---|---|---|---|---|---|
| AES EDK (CSP) | 128-256 bits | CKG;<br>DRBG;<br>AES-CBC;<br>AES-CFB128;<br>AES-ECB;<br>AES-CMAC;<br>AES-GCM;<br>AES-GMAC;<br>Cert: A4354 | Internally generated conformant to NIST SP 800-133r2 (CKG), section 6 for Symmetric Key Generation method, and the random value used in key generation is generated using SP 800-90Arev1 DRBG. | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Symmetric encryption/ decryption |
| AES KWK (CSP) | 128-256 bits | CKG;<br>DRBG;<br>AES-KW<br>Cert: A4354 | Internally generated conformant to NIST SP 800-133r2 (CKG), section 6 for Symmetric Key Generation method, and the random value used in key generation is generated using SP 800-90Arev1 DRBG. | Import: No<br>Export: No | N/A | Stored outside the module in the host OS | Zeroized via API command outside the module;<br>Power cycle | Key wrapping |
| RSA SGK (CSP) | 112-152 bits | CKG;<br>DRBG;<br>RSA KeyGen;<br>RSA SigVer<br>Cert: A4354 | Internally generated conformant to NIST SP 800-133r2 (CKG) using FIPS 186-4 RSA key generation method, and the random value used in the key generation is generated using SP 800-90Arev1 DRBG. | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Digital signature generation |
| RSA SVK (PSP[36]) | 112-152 bits | RSA SigVer;<br>Cert: A4354 | Internally derived conformant to FIPS 186-4 RSA key generation method | Import: No<br>Export: Via module API | MD/EE | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Digital signature verification |
| ECDSA SGK (CSP) | 128-192 bits | CKG;<br>DRBG;<br>ECDSA KeyGen;<br>ECDSA KeyVer;<br>ECDSA SigGen;<br>Cert: A4354 | Internally generated conformant to NIST SP 800-133r2 (CKG) using FIPS 186-4 ECDSA key generation method, and the random value used in key generation is generated using SP 800-90Arev1 DRBG | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Digital signature generation |
| ECDSA SVK (PSP) | 128-192 bits | ECDSA SigVer;<br>Cert: A4354 | Internally derived conformant to FIPS 186-4 ECDSA key generation method | Import: No<br>Export: Via module API | MD/EE | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Digital signature verification |
| DH public key (PSP) | 112-152 bits | KAS-FFC-SSC;<br>Cert: A4354 | Internally derived conformant to SP 800-56A rev3 DH key generation method | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent | Automatic zeroization when the tested platform is powered down | Key agreement |

[36] PSP – Public Security Parameter

| Keys/ SSP Name/Type | Strength | Security Function and Cert Number | Generation | Import/ Export | Establish-ment | Storage | Zeroization | Use and released keys |
|---|---|---|---|---|---|---|---|---|
| | | | | | | keys/SSPs storage | | |
| DH private key (CSP) | 112-152 bits | CKG;<br>DRBG;<br>KAS-FFC-SSC<br>Cert: A4354 | Internally generated conformant to NIST SP 800-133r2 (CKG) using SP 800-56Arev3 DH key generation method, and the random value used in the key generation is generated using SP 800-90Arev1 DRBG | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Key agreement |
| ECDH public key (PSP) | 128-256 bits | KAS-ECC-SSC;<br>Cert: A4354 | Internally derived conformant to SP 800-56A rev3 EC Diffie-Hellman key generation method | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Key agreement |
| ECDH private key (CSP) | 128-256 bits | CKG;<br>DRBG;<br>KAS-ECC-SSC;<br>Cert: A4354 | Internally generated conformant to NIST SP 800-133r2 (CKG) using SP 800-56Arev3 ECDH key generation method, and the random value used in the key generation is generated using SP 800-90Arev1 DRBG | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Key agreement |
| HMAC key (CSP) | 112 bits or greater | CKG;<br>DRBG;<br>HMAC-SHA-1;<br>HMAC-SHA2-256;<br>HMAC-SHA2-384<br>Cert: A4354 | Internally generated conformant to NIST SP 800-133r2 (CKG), section 6 for Symmetric Key Generation method, and the random value used in key generation is generated using SP 800-90Arev1 DRBG | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | MAC generation |
| DRBG entropy input (CSP) | 112 bits or greater | N/A | Obtained from the Entropy Source within TOEPP (GPS INT Pathways | Import: Via module API<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Random number generation |
| DRBG seed, (CSP) | 384 bits | CTR_DRBG<br>Cert: A4354 | Internally Derived from entropy input string as defined by NIST SP 800-90Arev1 | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Random number generation |
| DRBG internal state V value, (CSP) | 384 bits | CTR_DRBG<br>Cert: A4354 | Internally Derived from entropy input string as defined by NIST SP 800-90Arev1 | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Random number generation |
| DRBG key (CSP) | 384 bits | CTR_DRBG<br>Cert: A4354 | Internally Derived from entropy input string as defined by NIST SP 800-90Arev1 | Import: No<br>Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Random number generation |
| IKEv2 KDF secret (CSP) | 112-256 bits | KDF IKEv2<br>Cert: A4354 | Internally derived per the KDF defined in NIST SP 800-135 KDF (IKEv2) | Import: No<br>Export: | MD/EE | N/A: The module does not provide persistent | Automatic zeroization when the tested platform is powered down | Keying material used to derive other IPSec/IKEv2 keys |

| Keys/ SSP Name/Type | Strength | Security Function and Cert Number | Generation | Import/ Export | Establish-ment | Storage | Zeroization | Use and released keys |
|---|---|---|---|---|---|---|---|---|
| | | | | As part of agreement scheme | | keys/SSPs storage | | |
| SNMPv3 KDF secret (CSP) | 112-256 bits | KDF SNMP<br>Cert: A4354 | Internally derived per the KDF defined in NIST SP 800-135 KDF (SNMPv3) | Import: No<br>Export: As part of agreement scheme | MD/EE | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Keying material used to derive other SNMPv3 keys |
| SRTP KDF secret (CSP) | 112-256 bits | KDF SRTP<br>Cert: A4354 | Internally derived per the KDF defined in NIST SP 800-135 KDF (SRTP) | Import: No<br>Export: As part of agreement scheme | MD/EE | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Keying material used to derive other SRTP keys |
| SSH KDF secret (CSP) | 112-256 bits | KDF SSH<br>Cert: A4354 | Internally derived per the KDF defined in NIST SP 800-135 KDF (SSHv2) | Import: No<br>Export: As part of agreement scheme | MD/EE | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Keying material used to derive other SSHv2 keys |
| TLSv1.2 KDF extended master secret (CSP) | 112-256 bits | TLSv1.2 KDF with RFC 7627<br>Cert: A4354 | Internally derived per the KDF defined in NIST SP 800-135 KDF (TLSv1.2 with RFC 7627) | Import: No<br>Export: As part of agreement scheme | MD/EE | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Keying material used to derive other TLSv1.2 keys |
| TLSv1.3 KDF secret (CSP) | 112-256 bits | KDF TLSv1.3<br>Cert: A4354 | Internally derived per the KDF defined in NIST SP 800-135 KDF (TLSv1.3 with RFC 8446) | Import: No<br>Export: As part of agreement scheme | MD/EE | N/A: The module does not provide persistent keys/SSPs storage | Automatic zeroization when the tested platform is powered down | Keying material used to derive other TLSv1.3 keys |

## RBG entropy source

Table 11 below specifies the modules entropy sources.

**Table 11 - Non-Deterministic Random Number Generation Specification**

| Entropy sources | Minimum number of bits of entropy | Details |
|---|---|---|
| The OS passively loads entropy within the TOEPP into the module to seed the NIST SP 800-90Arev1 DRBG | At least 112 bits | While in the approved mode of operation, the entropy and seeding material for the NIST SP 800-90Arev1 DRBG are provided by the external calling application (and not by the module) which is outside the module's cryptographic boundary but contained within the module's physical perimeter. The module receives a LOAD command with entropy obtained from the entropy source inside the TOEPP. The minimum effective strength of the NIST SP 800-90Arev1 DRBG seed is required to be at least 112-bits when used in an approved mode of operation; therefore the minimum number of bits of entropy requested when the Module makes a call to the NIST SP 800-90Arev1 DRBG is at least 112-bits.<br>Per the IG 9.3.A Entropy Caveats, the following caveat applies: *No assurance of the minimum strength of generated SSPs (e.g., keys).* |

### Random Number Generation

The Approved DRBG for random number generation is a NIST SP 800-90Arev1 CTR_DRBG using AES-256 with derivation function and without prediction resistance. The numbers used for key generation are all generated by the CTR_DRBG within the module. Per NIST SP 800-90Arev1, section 10.2.1.1, the internal state is the values of *V* and *Key*. Refer to Table 4 above for the CAVP certificate of the validated DRBG algorithm.

### SSP Generation

The module generates RSA, ECDSA, ECDH, and DH asymmetric key pairs compliant with FIPS 186-4, using a NIST SP 800-90Arev1 CTR_DRBG for random number generation. In accordance with FIPS 140-3 IG D.H, the cryptographic module performs CKG for asymmetric keys as per section 5 of NIST SP 800-133rev2 (vendor affirmed) by obtaining a random bit string directly from an approved DRBG. The random bit string supports the required security strength requested by the calling application (without any V, as described in Additional Comments 2 of IG D.H.).

The module generates AES symmetric keys compliant with FIPS PUB 197 and HMAC key compliant with FIPS PUB 198. All symmetric key generation is performed using a NIST SP 800-90Arev1 CRT_DRBG for rando number generation. In accordance with FIPS 140-3 IG D.H, the cryptographic module performs CKG for symmetric keys as per section 6 of NIST SP 800-133rev2.

### SSP Entry and Output

The module does not support manual SSP entry or intermediate key generation output. SSPs are not output through physical ports on the TOEPP. SSPs are input in plaintext form via API from the calling application to the module. SSPs are output in plaintext form from the module to the calling application within TOEPP.

Per ISO/IEC 19790, section 7.9.5, the module performs two independent internal actions to prevent the inadvertent output of sensitive information:

1. The module internally requests the random number generation service, confirming it executes successfully.
2. Once keys are generated, the module performs the PCT to verify the keys are correctly generated.

Once both actions are successfully completed, the module outputs the SSP to the calling application via the output API.

### SSP Storage

The module does not provide persistent storage of SSPs. SSP storage is performed by the tested platform.

### Zeroization

The module does not possess persistent storage of SSPs. The SSP value only exists in volatile memory of the host appliance and that value vanishes when the module is powered off. The procedure for secure sanitization of the module at the end of life is simply to power off the tested platform.

# 10 Self-tests

The module performs Pre-Operational Self-Tests and CASTs[37] before entering an approved mode of operation. The module is single threaded and will not return to the calling application until all self-tests are complete. If any of the pre-operational self-tests or conditional cryptographic algorithm self-tests fail, the module enters a critical error state and sends an error to the OS. The module supports two Error states, critical error state and soft error state. Following is an example of the error message displayed on the console of the host appliance in a critical error state:

*%CRYPTO-0-SELF_TEST_FAILURE: Encryption self-test failed (<failing test description>)*

In a critical error state no cryptographic operations are performed and data output is prohibited. The CO can clear the error state by restarting the module.

If a PCT fails, the module enters a soft error state, deletes the key, logs an error, and returns to the approved mode of operation. In the approved mode of operation the service may be retried or a new service may be performed. Following is an example of the error message displayed on the console of the host appliance in a soft error state:

*%CRYPTO-3-RSA_SELFTEST_FAILED: Generated RSA key failed self test*

If the module fails to retrieve enough entropy, the module enters a soft error state. The module deletes the DRBG value, then reseeds and reinitializes the DRBG.

**Pre-operational self-tests:**

Pre-operational firmware integrity test:

- HMAC-SHA2-256 KAT[38]
- Firmware integrity test (using HMAC-SHA2-256)

**Conditional self-tests:**

Conditional cryptographic algorithm tests (CASTs):

- AES ECB (128-bit) Encrypt KAT
- AES ECB (128-bit) Decrypt KAT
- AES-GCM (256-bit) Encrypt KAT
- AES-GCM (256-bit) Decrypt KAT
- AES GMAC (256-bit) KAT
- CTR_DRBG Instantiate KAT
- CTR_DRBG Generate KAT
- CTR_DRBG Reseed KAT
  Note: DRBG Health Tests as specified in NIST SP 800-90Arev1 Section 11.3 are performed.
- HMAC-SHA2-256 KAT
- SHA-1 KAT
- SHA2-256 KAT
- SHA2-512 KAT

[37] CAST - Cryptographic Algorithm Self-Tests

[38] KAT – Known Answer Test

- ECDSA P-256 with SHA2-256 SigGen KAT
- ECDSA P-256 with SHA2-256 SigVer KAT
- RSA 2048-bit modulus with SHA2-256 SigGen KAT
- RSA 2048-bit modulus with SHA2-256 SigVer KAT
- KAS-FFC-SSC (shared secret computation) using MODP-2048 Primitive ‘Z’ KAT
- KAS-ECC-SSC (shared secret computation) using P-256 Primitive ‘Z’ KAT
- KDF IKEv2 KAT
- KDF SNMP KAT
- KDF SRTP KAT
- KDF SSH KAT
- TLSv1.2 KDF with RFC7627 KAT
- TLSv1.3 KDF KAT

Conditional pair-wise consistency tests (PCT):

- ECDSA PCT
- RSA PCT

**Periodic/Self-Tests On-Demand**

In addition to the automatic execution of self-tests at cryptographic module initialization, the CO can manually initiate self-tests on demand by executing the “test crypto self-test” command through the console of the host appliance. This command calls the “crypto_engine_nist_run_self_tests() function”. Self-tests can be executed on demand by power-cycling the module.

## 11 Life-cycle assurance

The module meets all the Level 1 requirements for FIPS 140-3. The module is completely and permanently embedded into Host Device IOS-XE OS. There are no installation considerations besides the loading of the IOS-XE OS. The module cannot be modified, replaced, or upgraded except by loading a new Host Device IOS-XE version in its entirety.

The module functions entirely within the process space of the process that invokes it, and thus satisfies the FIPS 140-3 requirement for a single user mode of operation.

During system start-up, the IOS-XE OS will call module’s ic2m_init() function. The ic2m_init() function is the default entry point for the module. The ic2m_init() function initiates all self-tests and does not return to the IOS-XE OS until all self-tests are completed successfully and the module is in an approved mode of operation. No other tasks are executed while the self-tests are performed so no data is passed and all cryptographic operations are prohibited. If a self-test fails, the module enters a critical error state and must be reloaded to clear the error state and retry the self-tests.

**AES-GCM IV:**

The CO shall consider the following requirements and restrictions when using the module.

- The AES-GCM IV is constructed in compliance with IG C.H, scenario 1 (TLS v1.2), scenario 2 (IPsec-v3), and scenario 5 (TLS v1.3). Users should consult IG C.H specific scenarios for all the requirements using AES-GCM mode. The TLS and IPsec/IKE protocols have not been reviewed or tested by the CAVP and CMVP.

- The AES-GCM IV generation follows RFC 5288 and shall only be used for the TLS protocol v1.2. The counter portion of the IV is set by the module within its cryptographic boundary. The module does not implement the TLS protocol. The module's implementation of AES-GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS protocol implicitly ensures that the nonce_explicit, or counter portion of the IV will not exhaust all of its possible values.
- The AES-GCM IV generation follows RFC 4106 and shall only be used for the IPsec-v3 protocol version 3. The counter portion of the IV is set by the module within its cryptographic boundary. The module does not implement the IPsec protocol. The module's implementation of AES-GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the IPsec protocol implicitly ensures that the nonce_explicit, or counter portion of the IV will not exhaust all of its possible values.
- The AES-GCM IV generation follows RFC 8446 and shall only be used for the TLS protocol v1.3. The counter portion of the IV is set by the module within its cryptographic boundary. The module does not implement the TLS protocol. The module's implementation of AES-GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS protocol implicitly ensures that the nonce_explicit, or counter portion of the IV will not exhaust all of its possible values.
- In these protocols if the module's power is lost and then restored, the key used for the AES GCM encryption/decryption shall be re-distributed. This condition is not enforced by the module; however, it is met implicitly. The module does not retain any state when power is lost. The AES-GCM key/IVs are not persistently stored during power off: therefore, there is no re-connection possible when the power is restored with re-generation of the key used for AES-GCM. After restoration of the power, the user of the module (e.g., TLS, IKE) along with User application that implements the protocol, must perform a complete new key establishment operation using new random numbers (Entropy input string, DRBG seed, DRBG internal state V and Key, shared secret values that are not retained during power cycle) and subsequent KDF operations to establish a new AES-GCM key/IV pair on either side of the network communication channel.

## 12 Mitigation of other attacks

The module does not support mitigation of other attacks as defined under ISO/IEC 19790, section 7.12.

# Appendix A – Acronyms and Terms

Table 12 below provides a list of acronyms and terms used throughout this Security Policy.

**Table 12 - Acronyms and Terms**

| Term/Acronym | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| API | Application Programming Interface |
| ASR | Aggregated Services Router |
| CAST | Cryptographic Algorithm Self-Tests |
| CAVP | Cryptographic Algorithm Validation Program |
| CBC | Cipher-Block Chaining |
| CFB | Cipher Feedback |
| CKG | Cryptographic Key Generation |
| CMAC | Cipher-based Message Authentication Code |
| CMVP | Cryptographic Module Validation Program |
| CO | Crypto Officer |
| CSP | Critical Security Parameter |
| CTR | Counter |
| CVL | Component Validation List |
| DH | Diffie-Hellman |
| DRBG | Deterministic Random Bit Generator |
| ECB | Electronic Codebook |
| ECC | Elliptic Curve Cryptography |
| ECDH | Elliptic Curve Diffie-Hellman |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| FFC | Finite Field Cryptography |
| FIPS | Federal Information Processing Standards |
| GCM | Galois Counter Mode |
| GMAC | Galois Message Authentication Code |
| HMAC | (keyed)-Hashed Message Authentication Code |
| IEC | International Electrotechnical Commission |
| IG | Implementation Guidance |
| IKE | Internet Key Exchange |
| IPsec | IP Security |
| ISO | International Organization of Standardization |
| KAS | Key Agreement Scheme |
| KAT | Known Answer Test |
| KDF | Key Derivation Function |
| N/A | Not Applicable |
| NIST | National Institute of Standards and Technology |
| OS | Operating System |
| PCT | Pairwise Consistency Test |
| PSP | Public Security Parameter |
| PUB | Publication |
| RFC | Request for Comment |
| RSA | Rivest, Shamir, and Adleman |
| SHA | Secure Hash Algorithm |
| SHS | Secure Hash Standard |
| SSC | Shared Secret Computation |
| SSH | Secure Shell |
| SNMP | Simple Network Transfer Protocol |
| SP | Special |

| Term/Acronym | Definition |
|---|---|
| SRTP | Secure Real-time Protocol |
| SSP | Sensitive Security Parameters |
| TLS | Transport Layer Security |
| TOEPP | Tested Operational Environment’s Physical Perimeter |