# ISO/IEC 19790 and FIPS 140-3 Non-Proprietary

# Security Policy

# for

# Firepower Management Center Virtual VMware Cryptographic Module

## Last Updated:  June 14, 2024, Version 0.3

# Table of Content

List of Figures

List of Tables

# 1 General

This is Cisco Systems, Inc. non-proprietary security policy for Firepower Management Center Virtual VMware Cryptographic Module (hereinafter referred to as the Module or FMCv) with software version 7.0.5. 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 Software 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. Table 1 below indicates the actual security levels for each area of the cryptographic module.

| ISO/IEC 24759:2017 Section 6 | ISO/IEC 24759:2017 and FIPS 140-3 Section Title | 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 | N/A |
| 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 |

**Table 1 Security Levels**

The module has an overall security level of 1.

# 2 Cryptographic module specification

The Module is a multi-chip standalone software module deployed as the virtualized version of the Cisco Firepower Management Center with underlying operating system identified as Linux 4 (also referred to as **F**irepower e**X**tensible **O**perating **S**ystem or FX-OS) throughout this document. The module's operational environment is non-modifiable.

The module is the administrative nerve center for managing critical Cisco network security solutions. It provides complete and unified management over firewalls, application control, intrusion prevention, URL filtering, and advanced malware protection, quickly and easily go from managing a firewall to controlling applications to investigating and remediating malware outbreaks. It is a key part of the broad and integrated Cisco Secure portfolio, delivering in-depth analysis, streamlined security management across the network and cloud, and accelerated incident investigation and response, working across Cisco and third-party technologies. The Firewall Management Center (FMC) discovers real-time information about changing network resources and operations. The Management Center is the centralized point for event and policy management for the following solutions:

- Cisco Firepower Next-Generation Firewall (NGFW)
- Cisco ASA with FirePOWER Services
- Cisco Firepower Next-Generation IPS (NGIPS)
- Cisco FirePOWER Threat Defense

- Cisco Advanced Malware Protection (AMP)

The module has been tested on the following Operational Environments.

| # | Operating System | Hardware Platform | Processor | PAA/Acceleration |
|---|---|---|---|---|
| 1 | Linux 4 (FX-OS) on VMware ESXi 6.7 | UCS C220 M5 SFF Server | Intel Xeon Gold 6128 (Skylake) | With PAA |
| 2 | Linux 4 (FX-OS) on VMware ESXi 6.7 | UCS C220 M5 SFF Server | Intel Xeon Gold 6128 (Skylake) | Without PAA |
| 3 | Linux 4 (FX-OS) on VMware ESXi 7.0 | UCS C220 M5 SFF Server | Intel Xeon Gold 6128 (Skylake) | With PAA |
| 4 | Linux 4 (FX-OS) on VMware ESXi 7.0 | UCS C220 M5 SFF Server | Intel Xeon Gold 6128 (Skylake) | Without PAA |

**Table 2  Tested Operational Environment**



**Figure 1  UCS C220 M5 Front view without Bezel (top) and with Bezel (bottom)[1]**



**Figure 2  UCS C220 M5 Rear view**

In addition to the platforms listed in Table 2, Cisco has also tested the module on the following platforms and claims vendor affirmation on them.

| # | Operating System | Hardware Platform |
|---|---|---|
| 1 | Linux 4 (FX-OS) | C220 M5 w/KVM/AWS |
| 2 | Linux 4 (FX-OS) | C240 M5 w/ESXi/KVM/AWS |
| 3 | Linux 4 (FX-OS) | C480 M5 w/ESXi/KVM/AWS |
| 4 | Linux 4 (FX-OS) | E160-M3 w/ESXi/KVM/AWS |
| 5 | Linux 4 (FX-OS) | E180D-M3 w/ESXi/KVM/AWS |

**Table 3  Vendor Affirmed Operational Environments**

---

[1] https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/c220m5-sff-specsheet.pdf

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.

**Mode of operation**

The module has one approved mode of operation and is always in the approved mode of operation after initial operations are performed (See Section 11). The module does not claim implementation of a degraded mode of operation. Section 4 provides details on the service indicator implemented by the module.

The table below lists 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. There are some algorithm modes that were tested but not implemented by the module. Only the algorithms, modes, and key sizes that are implemented by the module are shown in this table.

| CAVP Cert | Algorithm and Standard | Mode/Method | Description / Key Size(s) / Key Strength(s) | Use/Function |
|---|---|---|---|---|
| A2952 and A3376 | AES [FIPS 197; SP800-38A] | CBC | Key Length: 128 and 256 bits | Symmetric Encryption and Decryption. |
| A2952 and A3376 | AES [FIPS 197; SP 800-38D] | GCM | Key Length: 128 and 256 bits | Authenticated Symmetric Encryption and Decryption |
| A2952 and A3376 | KDF SSH [SP 800-135rev1] (CVL) | KDF SSH | N/A | Key derivation function used in SSHv2 |
| A2952 and A3376 | TLS v1.2 KDF RFC7627 [RFC7627] (CVL) | TLS v1.2 KDF RFC7627 | N/A | Key derivation in TLSv1.2 with RFC7627 KDF with Extended Master Secret |
| A2952 and A3376 | CTR_DRBG [SP 800-90Arev1] | AES-256 Derivation Function Enabled; Prediction Resistance: Yes | N/A | Random number generation |
| A2952 and A3376 | ECDSA [FIPS 186-4] | ECDSA KeyGen | Curves: P-256, P-384, P-521 | ECDSA keypair generation |
| A2952 and A3376 | ECDSA [FIPS 186-4] | ECDSA KeyVer | Curves: P-256, P-384, P-521 | ECDSA keypair verification |
| A2952 and A3376 | ECDSA [FIPS 186-4] | ECDSA SigGen | Curves: P-256, P-384, P-521 | ECDSA signature generation |
| A2952 and A3376 | ECDSA [FIPS 186-4] | ECDSA SigVer | Curves: P-256, P-384, P-521 | ECDSA Signature verification |
| A2952 and A3376 | HMAC [FIPS 198-1] | HMAC-SHA-1 | Key Length: 112 bits or greater | Keyed Hash |
| A2952 and A3376 | HMAC [FIPS 198-1] | HMAC-SHA2-256 | Key Length: 112 bits or greater | Keyed Hash |
| A2952 and A3376 | HMAC [FIPS 198-1] | HMAC-SHA2-384 | Key Length: 112 bits or greater | Keyed Hash |
| A2952 and A3376 | HMAC [FIPS 198-1] | HMAC-SHA2-512 | Key Length: 112 bits or greater | Keyed Hash |
| A2952 and A3376 | KAS-SSC [SP 800-56Arev3] | KAS-ECC-SSC: Scheme: ephemeralUnified: KAS Role: initiator, responder | Curves: P-256, P-384, P-521; Key establishment methodology provides between 128 and 256 | KAS-ECC shared secret computation |

| CAVP Cert | Algorithm and Standard | Mode/Method | Description / Key Size(s) / Key Strength(s) | Use/Function |
|---|---|---|---|---|
| | | | bits of encryption strength | |
| A2952 and A3376 | KAS [SP 800-56Arev3] | KAS (ECC): Scheme: ephemeralUnified KAS Role: initiator, responder<br><br>KAS-SSC Cert. #A2952, TLSv1.2 KDF RFC7627 Cert. #A2952;<br><br>KAS-SSC Cert. #A3376, TLSv1.2 KDF RFC7627 Cert. #A3376 | Curves: P-256, P-384 and P-521;<br><br>Key establishment methodology provides between 128 and 256 bits of encryption strength | Key Agreement Scheme per SP800-56Arev3 with key derivation function (SP800-135rev1)<br><br>Note: The module's KAS (ECC) implementation is FIPS140-3 IG D.F Scenario 2 (path 2) compliant |
| A2952 and A3376 | KAS-SSC [SP 800-56Arev3] | KAS-FFC-SSC: Scheme: dhEphem: KAS Role: initiator, responder | MODP-2048;<br><br>Key establishment methodology provides between 112 bits of encryption strength | KAS-FFC shared secret computation |
| A2952 and A3376 | KAS [SP 800-56Arev3] | KAS (FFC): Scheme: dhEphem KAS Role: initiator, responder<br><br>KAS-SSC #A2952, KDF SSH Cert. #A2952;<br><br>KAS-SSC Cert. #A3376, KDF SSH Cert. #A3376 | MODP-2048;<br><br>Key establishment methodology provides between 112 bits of encryption strength | Key Agreement Scheme per SP800-56Arev3 with key derivation function (SP800-135rev1)<br><br>Note: The module's KAS (FFC) implementation is FIPS140-3 IG D.F Scenario 2 (path 2) compliant |
| A2952 and A3376 | RSA [FIPS 186-4] | RSA KeyGen: - Mode: B.3.4 - 2048/3072 Modulus | Modulus: 2048/3072 bits | RSA keypair generation |
| A2952 and A3376 | RSA [FIPS 186-4] | RSA SigGen: - PKCSv1.5 - 2048/3072 Modulus with SHA2-256/384/512 | Modulus: 2048/3072 bits | RSA signature generation |
| A2952 and A3376 | RSA [FIPS 186-4] | RSA SigVer: - PKCSv1.5 - 2048/3072 Modulus with SHA2-256/384/512 | Modulus: 2048/3072 bits | RSA signature verification |
| A2952 and A3376 | Safe Primes Key Generation [SP 800-56Arev3] | KeyGen for KAS-SSC (FFC) | Safe Prime Groups: MODP-2048 | KAS-FFC Keypair domain parameters generation |
| A2952 and A3376 | SHS [FIPS 180-4] | SHA-1 | N/A | Message Digest |
| A2952 and A3376 | SHS [FIPS 180-4] | SHA2-256 | N/A | Message Digest |
| A2952 and A3376 | SHS [FIPS 180-4] | SHA2-384 | N/A | Message Digest |
| A2952 and A3376 | SHS [FIPS 180-4] | SHA2-512 | N/A | Message Digest |

| CAVP Cert | Algorithm and Standard | Mode/Method | Description / Key Size(s) / Key Strength(s) | Use/Function |
|---|---|---|---|---|
| Vendor Affirmed | CKG (SP800-133rev2) | Section 5.1, Section 5.2 | Cryptographic Key Generation; SP 800-133rev2 and IG D.H. | Key Generation.<br><br>Note: The cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per section 5 in SP800-133rev2 (vendor affirmed). A seed (i.e., the random value) used in asymmetric key generation is a direct output from SP800-90Arev1 CTR_DRBG. |

**Table 4  Approved Algorithms**

Notes:
- Algorithm Cert. #A2952 was tested for the OE with PAA
- Algorithm Cert. #A3376 was tested for the OE without PAA
- The module's AES-GCM implementation conforms to FIPS 140-3 IG C.H scenario #1 following RFC 5288 for TLS. The module is compatible with TLSv1.2 and provides support for the acceptable GCM cipher suites from SP 800-52 Rev1, Section 3.3.1. The operations of one of the two parties involved in the TLS key establishment scheme were performed entirely within the cryptographic boundary of the module being validated. 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.
- No parts of SSH and TLS protocols, other than the KDFs, have been tested by the CAVP and CMVP.
- In accordance with FIPS 140-3 IG D.H, the cryptographic module performs Cryptographic Key Generation as per section 5 in SP800-133rev2. The resulting generated seed used in the asymmetric key generation is the unmodified output from SP800-90Arev1 DRBG.

As the module can only be operated in the Approved mode of operation, and any algorithms not listed in the table 4 above will be rejected by the module while in the approved mode, the tables defined in SP800-140B for the following categories are missing from this document.
- Non-Approved Algorithms Allowed in Approved Mode of Operation
- Non-Approved Algorithms Allowed in Approved Mode of Operation with No Security Claimed
- Non-Approved Algorithms Not Allowed in Approved Mode of Operation

**Cryptographic boundary**

The module is defined as a multi-chip standalone software module (inside red dashed area). The cryptographic boundary includes all of the module's software components, including Guest OS, API and FOM Crypto Library (Cisco FIPS Object Module). The physical perimeter is the Tested Operational Environment's Physical Perimeter (TOEPP) on which the module runs.
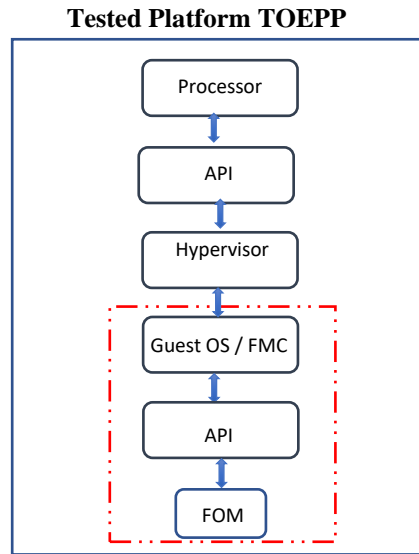
**Tested Platform TOEPP**



**Figure 3 Block Diagram**

Note: Block Diagram above comprises the following components:
- Processor = Chip on the tested platform to handle all processes.
- API = Host API between hypervisor and processor
- Hypervisor = VMWare ESXi 6.7 and 7.0
- Guest OS/FMC = Linux 4 (FX-OS)
- API = Guest API between the FMC Module and FOM Crypto library
- FOM = Cisco FIPS Object Module (FOM) Crypto Library

# 3 Cryptographic module 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 module's logical interfaces provided by the module are mapped onto the FIPS 140-3 logical interfaces (data input, data output, control input, control output and status output) as follows.

| Physical Port | Logical Interface | Data that passes over port/interface |
|---|---|---|
| N/A | Data Input Interface | Arguments for an API call that provide the data to be used or processed by the module. |
| N/A | Data Output Interface | Arguments output from an API call. |
| N/A | Control Input Interface | Arguments for an API call used to control and configure module operation. |
| N/A | Control Output Interface | N/A |
| N/A | Status Output Interface | Return values, and or log messages. |

**Table 5 Ports and Interfaces**

# 4 Roles, services, and authentication

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. The module provides the following services to the Crypto Officer.

| Role | Service | Input | Output |
|------|---------|-------|--------|
| Crypto Officer | Show Status | API command to show status | Module's current status |
| Crypto Officer | Show Version | API commands to show version | Module's name/ID and versioning information |
| Crypto Officer | Perform Self-Tests | API commands to conduct on-demand Self-Tests | Status of the self-tests results |
| Crypto Officer | Perform Zeroization | API commands to conduct Zeroization operation or Power down the tested platform | Status of the SSPs zeroization |
| Crypto Officer | Configure Network | API Commands to configure the module | Status of the completion of network related configuration |
| Crypto Officer | Configure SSHv2 Function | API commands to configure SSHv2 | Status of the completion of SSHv2 configuration |
| Crypto Officer | Configure HTTPS over TLSv1.2 Function | API commands to configure HTTPS over TLSv1.2 | Status of the completion of HTTPS over TLSv1.2 configuration |
| Crypto Officer | Run SSHv2 Function | API commands to execute SSHv2 service | Status of SSHv2 secure tunnel establishment |
| Crypto Officer | Run HTTPS over TLSv1.2 Function | API commands to execute HTTPS over TLSv1.2 service | Status of HTTPS over TLSv1.2 secure tunnel establishment |

**Table 6  Roles, Service Commands, Input and Output**

Table 7 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 = Zeroise**: The module zeroises the SSP.

**N/A** = The service does not access any SSP during its operation.

| Services | Description | Approved Security Functions | Keys and /or SSPs | Roles | Access rights to Keys and/or SSPs | Indicator |
|----------|-------------|------------------------------|-------------------|-------|-----------------------------------|-----------|
| Show Status | Provide Module's current status | N/A | N/A | Crypto Officer | N/A | None |
| Show Version | Provide Module's name/ID and versioning information | N/A | N/A | Crypto Officer | N/A | None |
| Perform Self-Tests | Perform Self-Tests (Pre-operational self-tests and Conditional Self-Tests) | N/A | N/A | Crypto Officer | N/A | None |
| Perform Zeroization | Perform Zeroization | N/A | All SSPs | Crypto Officer | Z | None |
| Configure Network | Sets configuration of the systems | N/A | N/A | Crypto Officer | N/A | None |
| Configure SSHv2 Function | Configure SSHv2 Function | AES-CBC; CKG; | Diffie-Hellman Private Key; | Crypto Officer | W, E | Global Indicator and SSHv2 |

| Services | Description | Approved Security Functions | Keys and /or SSPs | Roles | Access rights to Keys and/or SSPs | Indicator |
|---|---|---|---|---|---|---|
| | | KDF SSH; CTR_DRBG; HMAC-SHA-1; HMAC-SHA2-256; HMAC-SHA2-384; HMAC-SHA2-512; KAS-FFC-SSC; KAS (FFC); RSA KeyGen; RSA SigGen; RSA SigVer; Safe Primes Key Generation; SHA-1; SHA2-256; SHA2-384; SHA2-512 | Diffie-Hellman Public Key; Peer Diffie-Hellman Public Key; Diffie-Hellman Shared Secret; RSA Private Key; RSA Public Key; SSH Session Integrity Key; SSH Session Key | | | success log message |
| Configure HTTPS over TLSv1.2 Function | Configure HTTPS over TLSv1.2 Function | AES-CBC; AES-GCM; CKG; TLS v1.2 KDF RFC7627; CTR_DRBG; ECDSA KeyGen; ECDSA KeyVer; ECDSA SigGen; ECDSA SigVer; HMAC-SHA-1; HMAC-SHA2-256; HMAC-SHA2-384; HMAC-SHA2-512; KAS-ECC-SSC; KAS (ECC); RSA KeyGen; RSA SigGen; RSA SigVer; SHA-1; SHA2-256; SHA2-384; SHA2-512 | EC Diffie-Hellman Private Key; EC Diffie-Hellman Public Key; Peer EC Diffie-Hellman Public Key; EC Diffie-Hellman Shared Secret; ECDSA Private Key; ECDSA Public Key; RSA Private Key; RSA Public Key; TLS master secret; TLS Session Key; TLS Session Integrity Key | Crypto Officer | W, E | Global Indicator and HTTPS over TLSv1.2 success log message |
| Run SSHv2 Function | Execute SSHv2 Function | AES-CBC; CKG; KDF SSH; CTR_DRBG; HMAC-SHA-1; HMAC-SHA2-256; | DRBG entropy input; DRBG Seed, Internal State V value, and Key; Diffie-Hellman Private Key; | Crypto Officer | W, E | Global Indicator and SSHv2 success log message |

| Services | Description | Approved Security Functions | Keys and /or SSPs | Roles | Access rights to Keys and/or SSPs | Indicator |
|---|---|---|---|---|---|---|
| | | HMAC-SHA2-384; HMAC-SHA2-512; KAS-FFC-SSC; KAS (FFC); RSA KeyGen; RSA SigGen; RSA SigVer; Safe Primes Key Generation; SHA-1; SHA2-256; SHA2-384; SHA2-512 | Diffie-Hellman Public Key; Peer Diffie-Hellman Public Key; Diffie-Hellman Shared Secret; RSA Private Key; RSA Public Key; SSH Session Integrity Key; SSH Session Key | | | |
| Run HTTPS over TLSv1.2 Function | Execute HTTPS over TLSv1.2 Function | AES-CBC; CKG; TLS v1.2 KDF RFC7627; CTR_DRBG; ECDSA KeyGen; ECDSA KeyVer; ECDSA SigGen; ECDSA SigVer; HMAC-SHA-1; HMAC-SHA2-256; HMAC-SHA2-384; HMAC-SHA2-512; KAS-ECC-SSC; KAS (ECC); RSA KeyGen; RSA SigGen; RSA SigVer; SHA-1; SHA2-256; SHA2-384; SHA2-512 | DRBG entropy input; DRBG Seed, Internal State V value, and Key; EC Diffie-Hellman Private Key; EC Diffie-Hellman Public Key; Peer EC Diffie-Hellman Public Key; EC Diffie-Hellman Shared Secret; ECDSA Private Key; ECDSA Public Key; RSA Private Key; RSA Public Key; TLS master secret; TLS Session Key; TLS Session Integrity Key | Crypto Officer | W, E | Global Indicator and HTTPS over TLSv1.2 success log message |

**Table 7 Approved Services**

As the module can only be operated in the Approved mode of operation, as such any algorithms not listed in Table 4 above will be rejected by the module while in the approved mode, the table required defined in SP800-140B for Non-Approved Services is missing from this document.

The module doesn't support self-initiated cryptographic output capability and cryptographic Bypass capability services.

# 5 Software/Firmware security

**Integrity techniques**
The module is provided in the form of binary executable code. To ensure the software security, the module is protected by HMAC-SHA2-512 (HMAC Certs. #A2952 or #A3376) algorithm. The software integrity test key (non-SSP) was preloaded to the module's binary the factory and used for software integrity test only at the pre-operational self-test. 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 to an Error state with all crypto functionality inhibited.

**Integrity test on-demand**
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 software integrity test on-demand.

# 6 Operational environment

The module is a software module, which is operated in a modifiable operational environment per FIPS 140-3 level 1 specifications. The module's software version running on each tested platform is 7.0.5.

The module has control over its own SSPs. The process and memory management functionality of the host device's OS prevents 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

The FIPS 140-3 physical security requirements do not apply to the Module since it is a software module.

# 8 Non-invasive security

Currently, non-invasive security is not required by FIPS 140-3 (see NIST SP 800-140F).  The requirements of this area are not applicable to the module.

# 9 Sensitive security parameters management

The following table summarizes the keys and Sensitive Security Parameters (SSPs) that are used by the cryptographic services implemented in the module.

| Key/SSP Name Type | Strength | Security Function and Cert Number | Generation | Import/ Export | Esta blish ment | Storage | Zeroization | Use & related Keys |
|---|---|---|---|---|---|---|---|---|
| DRBG entropy input (CSP) | 384 bits | CTR_DRBG #A2952 or #A3376 | Obtained from the Entropy Source within | Import to the module via Module's API | N/A | N/A: The module does not provide | Automatic zeroization when the tested | Random Number Generation |

| Key/SSP Name Type | Strength | Security Function and Cert Number | Generation | Import/ Export | Esta blish ment | Storage | Zeroization | Use & related Keys |
|---|---|---|---|---|---|---|---|---|
| | | | TOEPP (GPS INT Pathways) | Export: No | | persistent keys/SSPs storage. | platform is powered down | |
| DRBG Seed, Internal State V value, and Key (CSP) | 256 bits | CTR_DRBG #A2952 or #A3376 | Internally Derived from entropy input string as defined by SP800-90Arev1 | Import: No 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 |
| Diffie-Hellman Private Key (CSP) | MODP-2048 | CKG; CTR_DRBG; KAS (FFC); KAS-FFC-SSC; Safe Primes Key Generation #A2952 or #A3376 | Internally generated conformant to SP800-133r2 (CKG) using SP800-56A rev3 Diffie-Hellman key generation method, and the random value used in key generation is generated using SP800-90ARev1 DRBG | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/ SSPs storage. | Automatic zeroization when the tested platform is powered down | Used to derive Diffie-Hellman Shared Secret |
| Diffie-Hellman Public Key (PSP) | MODP-2048 | KAS (FFC); KAS-FFC-SSC; Safe Primes Key Generation #A2952 or #A3376 | Internally derived per the Diffie-Hellman key agreement (SP800-56Arev3) | Import: No Export: to the SSH Peer application | N/A | N/A: The module does not provide persistent keys/ SSPs storage. | Automatic zeroization when the tested platform is powered down | Used to derive Diffie-Hellman Shared Secret |
| Peer Diffie-Hellman Public Key (PSP) | MODP-2048 | N/A | N/A | Import: to the Module via API Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when the tested platform is powered down | Used to derive Diffie-Hellman Shared Secret |
| Diffie-Hellman Shared Secret (CSP) | MODP-2048 | KAS-FFC-SSC #A2952 or #A3376 | Internally generated using SP800-56Arev3 DH shared secret computation | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when the tested platform is powered down | Used to derive SSH session related keys |

| Key/SSP Name Type | Strength | Security Function and Cert Number | Generation | Import/ Export | Esta blish ment | Storage | Zeroization | Use & related Keys |
|---|---|---|---|---|---|---|---|---|
| EC Diffie-Hellman Private Key (CSP) | P-256, P-384 and P-521 | CKG; CTR_DRBG; KAS (ECC); KAS-ECC-SSC #A2952 or #A3376 | Internally generated conformant to SP800-133r2 (CKG) using SP800-56A rev3 EC Diffie-Hellman key generation method, and the random value used in key generation is generated using SP800-90Arev1 DRBG | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/ SSPs storage. | Automatic zeroization when the tested platform is powered down | Used to derive EC Diffie-Hellman Shared Secret |
| EC Diffie-Hellman Public Key (PSP) | P-256, P-384 and P-521 | KAS (ECC); KAS-ECC-SSC #A2952 or #A3376 | Internally derived per the EC Diffie-Hellman key agreement (SP800-56Arev3) | Import: No Export: to the TLS Peer application | N/A | N/A: The module does not provide persistent keys/ SSPs storage. | Automatic zeroization when the tested platform is powered down | Used to derive EC Diffie-Hellman Shared Secret |
| Peer EC Diffie-Hellman Public Key (PSP) | P-256, P-384 and P-521 | N/A | N/A | Import: to the Module via API Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when the tested platform is powered down | Used to derive EC Diffie-Hellman Shared Secret |
| EC Diffie-Hellman Shared Secret (CSP) | P-256, P-384 and P-521 | KAS-ECC-SSC #A2952 or #A3376 | Internally generated using SP800-56Ar3 ECDH shared secret computation | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when the tested platform is powered down | Used to derive TLS session related keys |
| ECDSA Private Key (CSP) | P-256, P-384 and P-521 | CKG; CTR_DRBG; ECDSA KeyGen; ECDSA KeyVer; ECDSA SigGen; #A2952 or #A3376 | Internally generated conformant to SP800-133r2 (CKG) using FIPS 186-4 ECDSA key generation method, and the random value used in key generation is generated using SP800- | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when the tested platform is powered down | Signature generation and Verification used in TLS |

| Key/SSP Name Type | Strength | Security Function and Cert Number | Generation | Import/ Export | Establishment | Storage | Zeroization | Use & related Keys |
|---|---|---|---|---|---|---|---|---|
| | | | 90Arev1 DRBG | | | | | |
| ECDSA Public Key (PSP) | P-256, P-384 and P-521 | ECDSA KeyGen; ECDSA KeyVer; ECDSA SigVer; #A2952 or #A3376 | Internally derived per the FIPS 186-4 ECDSA key generation method | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when the tested platform is powered down | Signature generation and Verification used in TLS |
| RSA Private Key (CSP) | 2048 and 3072 bits | CKG; CTR_DRBG; RSA KeyGen; RSA SigGen; #A2952 or #A3376 | Internally generated conformant to SP800-133r2 (CKG) using FIPS 186-4 RSA key generation method, and the random value used in the key generation is generated using SP800-90Arev1 DRBG | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when the tested platform is powered down | Signature generation and Verification used in SSH or TLS |
| RSA Public Key (PSP) | 2048 and 3072 bits | KeyGen; RSA SigVer; #A2952 or #A3376 | Internally derived per the FIPS 186-4 RSA key generation method | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when the tested platform is powered down | Signature generation and Verification used in SSH or TLS |
| SSH Session Integrity Key (CSP) | 160 bits | KDF SSH; HMAC-SHA-1 #A2952 or #A3376 | Internally Derived per the key derivation function defined in SP800-135 KDF (KDF-SSH). | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when SSH session is terminated or when the tested platform is powered down | Used for SSH session integrity protection. |
| SSH Session Key (CSP) | 128/256 bits | AES-CBC; KDF SSH; #A2952 or #A3376 | Internally Generated via key derivation function defined in SP800-135 KDF (KDF-SSH) | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when SSH session is terminated or when the tested platform is | Used for SSH session confidentiality protection |

| Key/SSP Name Type | Strength | Security Function and Cert Number | Generation | Import/ Export | Esta blish ment | Storage | Zeroization | Use & related Keys |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | powered down | |
| TLS Master Secret (CSP) | 48 Bytes | Keying Material | Internally Derived per the key derivation function defined in SP800-135 KDF (KDF-TLS v1.2 RFC7627) | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when TLS session is terminated or when the tested platform is powered down | Keying material used to derive other TLS keys |
| TLS Session Key (CSP) | 128/256 bits | AES-CBC; AES-GCM; TLS v1.2 KDF RFC7627; #A2952 or #A3376 | Internally Derived per the key derivation function defined in SP800-135 KDF (KDF-TLS v1.2 RFC7627) | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when TLS session is terminated or when the tested platform is powered down | Used for TLS session confidentiality protection |
| TLS Session Integrity Key (CSP) | 256-384 bits | TLS v1.2 KDF RFC7627; HMAC-SHA2-256; HMAC-SHA2-384; #A2952 or #A3376 | Internally Derived per the key derivation function defined in SP800-135 KDF (KDF-TLS v1.2 RFC7627) | Import: No Export: No | N/A | N/A: The module does not provide persistent keys/SSPs storage. | Automatic zeroization when TLS session is terminated or when the tested platform is powered down | Used for TLS session integrity protection |

**Table 8  SSPs**

**RBG entropy source**

| Entropy sources | Minimum number of bits of entropy | Details |
|---|---|---|
| Entropy within the TOEPP was passively Load Into the module to seed the 800-90Arev1 DRBG by the Operating System. | At least 112 bits | While operating in the approved mode, the entropy and seeding material for the SP800-90Arev1 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 Tested Operational Environment's Physical Perimeter (TOEPP) boundary. The module receives a LOAD command with entropy obtained from the entropy source (Intel CPU processor with instructions RDRand) inside the TOEPP. The minimum effective strength of the 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 SP 800-90Arev1 DRBG is at least 112 bits.<br><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).* |

**Table 9 Non-Deterministic Random Number Generation Specification**

# 10 Self-tests

When the module is loaded or instantiated (after being powered off, rebooted, etc.), the module runs pre-operational self-tests. The operating system is responsible for the initialization process and loading of the Module. The module is designed with a default entry point (DEP) which ensures that the self-tests are initiated automatically when the module is loaded. Prior to the module providing any data output via the data output interface, the module would perform and pass the pre-operational self-tests. A software integrity test is performed on the runtime image of the module with HMAC-SHA2-512 algorithm. Prior to the firmware integrity test, the module conducts a HMAC-SHA2-512 Cryptographic Algorithm Self-test (CAST). If the CAST on the HMAC-SHA2-512 is successful, the HMAC value of the runtime image is recalculated and compared with the stored HMAC value pre-computed at compilation time. Following the successful pre-operational self-tests, the module would execute the Conditional Cryptographic Algorithm Self-tests (CASTs) for all approved cryptographic algorithms implemented by the module during power-up as well.

The self-test success or failure messages (e.g., *Error: Signature RSA test failure* or *ECDH P-256 test failure)* were logged, which is functioning as the self-test status indicator. If any one of the self-tests fails, the module transitions into an error state and outputs the error message via the module's status output interface. While the module is in the error state, all data through the data output interface and all cryptographic operations are disabled. The only method to recover from the error state is to power cycle the device which results in the module being reloaded into memory and reperforming the pre-operational software integrity test and the Conditional CASTs. The module will only enter into the operational state after successfully passing the preoperational software integrity test and the Conditional CASTs.

Below are the details of the self-tests conducted by the module.

**Pre-operational self-tests**:
- Pre-operational software integrity test
  - HMAC-SHA-512 KAT
  - Software Integrity Test (using HMAC-SHA-512)

Please note that the module conducts HMAC-SHA-512 KAT self-test before the integrity test is performed.

**Conditional self-test**

- Conditional cryptographic algorithm self-tests (CASTs)
  - AES-CBC 256 bits Encrypt KAT
  - AES-CBC 256 bits Decrypt KAT
  - AES-GCM 256 bits Authenticated Encrypt KAT
  - AES-GCM 256 bits Authenticated Decrypt KAT
  - CTR_DRBG Instantiate KAT
  - CTR_DRBG Generate KAT
  - CTR_DRBG Reseed KAT
    Note: CTR_DRBG Health Tests: Generate, Reseed, Instantiate functions per Section 11.3 of SP 800-90Arev1
  - ECDSA P-256 with SHA-256 SigGen KAT
  - ECDSA P-256 with SHA-256 SigVer KAT
  - HMAC-SHA-1 KAT
  - HMAC-SHA-256 KAT
  - HMAC-SHA-384 KAT
  - HMAC-SHA-512 KAT
  - KAS-ECC-SSC Primitive Z KAT
  - KAS-ECC-SSC Primitive Z KAT
  - RSA 2048 bits modulus with SHA-256 SigGen KAT
  - RSA 2048 bits modulus with SHA-256 SigVer KAT
  - SHA-1 KAT
  - KDF-SSH KAT
  - KDF-TLS KAT

The module generates RSA, ECDSA, KAS-ECC and KAS-FFC asymmetric keys and performs all required pair-wise consistency tests on the newly generated key pairs as detailed below.

- Conditional pair-wise consistency tests (PCTs)
  - RSA PCT
  - ECDSA PCT
  - KAS-ECC PCT
  - KAS-FFC PCT

**Periodic/Self-tests on-demand**

The module performs on-demand self-tests initiated by the operator, by power-cycling or rebooting the tested platform. The full suite of self-tests is then executed. The same procedure may be employed by the operator to perform periodic self-tests. In addition, it is recommended for the Crypto Officer to perform the periodic tests a minimum of once every 60 days to ensure all components are functioning correctly.

# 11 Life-cycle assurance

## 11.1 Secure operations

The module meets all the Level 1 requirements for FIPS 140-3. The validated Module's executable file Cisco_Firepower_Mgmt_Center_Virtual300_VMware-7.0.5-72-disk1.vmdk is the only allowable software image file running on the respective test platform listed in the Table 2 above while in the approved mode. The Crypto Officer must configure and enforce the following initialization steps:

**Step 1:** For all Management Centers, the setup process must be completed by logging into the Management Center's web interface and specifying initial configuration options on a setup page.

**Step 2:** Choose System > Configuration (Choose **SSH** or **HTTPS** or a combination of these options to specify which ports you want to enable for these IP addresses).

**Step 3:** System>Licenses>Smart Licenses, add and verify licenses (*Firepower Management Center Configuration Guide provides more detailed information).*

Install AES SMART license to use AES (for data traffic and SSH).

**Step 4:** System > Configuration; Devices > Platform Settings; STIG Compliance, choose Enable STIG Compliance; Click on save. This sets the approved mode of operation. The CO shall only use approved cryptographic algorithms listed in Section 2.2 above.

**Step 5**: Reboot the security appliances.

## 12  Mitigation of other attacks

The requirements under INCITS+ISO+IEC 19790+2012[2014], section 7.12 "Mitigation of other attacks", are not applicable to the module since the module currently doesn't support any mitigation of other attacks services.