# Enterprise Secure Key Manager

Hardware P/N M6H81AA, Version 5.0;
Firmware Version: 7.0.1

# FIPS 140-2
# Non-Proprietary Security Policy

**Level 2 Validation**

Document Version 1.0

**December 7, 2017**

On November 5, 2018, the Atalla business was acquired by Utimaco Inc. For aspects of this Security Policy document, the rest of this document will refer to the Enterprise Secure Key Manager. However, the Vendor is now Utimaco Inc.

# Table of Contents

# Table of Figures

# Table of Tables

# 1 Introduction

## 1.1 Purpose

This document is a non-proprietary Cryptographic Module Security Policy for the Enterprise Secure Key Manager (ESKM) from Micro Focus. Federal Information Processing Standards (FIPS) 140-2, *Security Requirements for Cryptographic Modules*, specifies the U.S. and Canadian Governments' requirements for cryptographic modules. The following pages describe how the ESKM meets these requirements and how to use the ESKM in a mode of operation compliant with FIPS 140-2. This policy was prepared as part of the Level 2 FIPS 140-2 validation of the Enterprise Secure Key Manager.

More information about FIPS 140-2 and the Cryptographic Module Validation Program (CMVP) is available at the website of the National Institute of Standards and Technology (NIST): http://csrc.nist.gov/groups/STM/cmvp/index.html.

In this document, the Enterprise Secure Key Manager is referred to as the *ESKM*, the *module*, or the *device*.

## 1.2 References

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

- The Micro Focus website (https://software.microfocus.com/en-us/home) contains information on the full line of products from Micro Focus.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/index.html) contains contact information for answers to technical or sales-related questions for the module.

# 2 Enterprise Secure Key Manager

## 2.1 Overview

Micro Focus provides a range of security products for banking, the Internet, and enterprise security applications. These products use encryption technology—often embedded in hardware—to safeguard sensitive data, such as financial transactions over private and public networks and to offload security processing from the server.

The Enterprise Secure Key Manager is a hardened server that provides security policy and key management services to encrypting client devices and applications. After enrollment, clients, such as storage systems, application servers and databases, make requests to the ESKM for creation and management of cryptographic keys and related metadata.

Client applications can access the ESKM via its Key Management Service (KMS) server and the Key Management Interoperability Protocol (KMIP) server. Configuration and management can be performed via web administration, Secure Shell (SSH), or serial console. Status-monitoring interfaces include a dedicated FIPS status interface, a health check interface, and Simple Network Management Protocol (SNMP).

The deployment architecture of the Enterprise Secure Key Manager is shown in Figure 1 below.

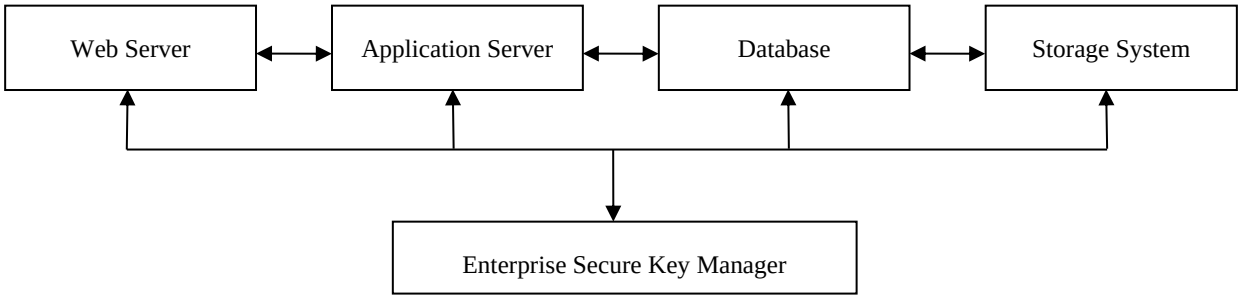


**Figure 1 – Deployment Architecture of the Enterprise Secure Key Manager**

## 2.2 Cryptographic Module Specification

The Enterprise Secure Key Manager is validated at FIPS 140-2 section levels shown in Table 1.

**Table 1 – Security Level per FIPS 140-2 Section**

| Section | Section Title | Level |
|---|---|---|
| 1 | Cryptographic Module Specification | 2 |
| 2 | Cryptographic Module Ports and Interfaces | 2 |
| 3 | Roles, Services, and Authentication | 3 |
| 4 | Finite State Model | 2 |
| 5 | Physical Security | 2 |
| 6 | Operational Environment | N/A |
| 7 | Cryptographic Key Management | 2 |

| Section | Section Title | Level |
|---|---|---|
| 8 | EMI/EMC | 2 |
| 9 | Self-Tests | 2 |
| 10 | Design Assurance | 2 |
| 11 | Mitigation of Other Attacks | N/A |

The block diagram of the module is given in Figure 2. The cryptographic boundary is clearly shown in the figure. Notice that the power supplies are not included in the boundary.

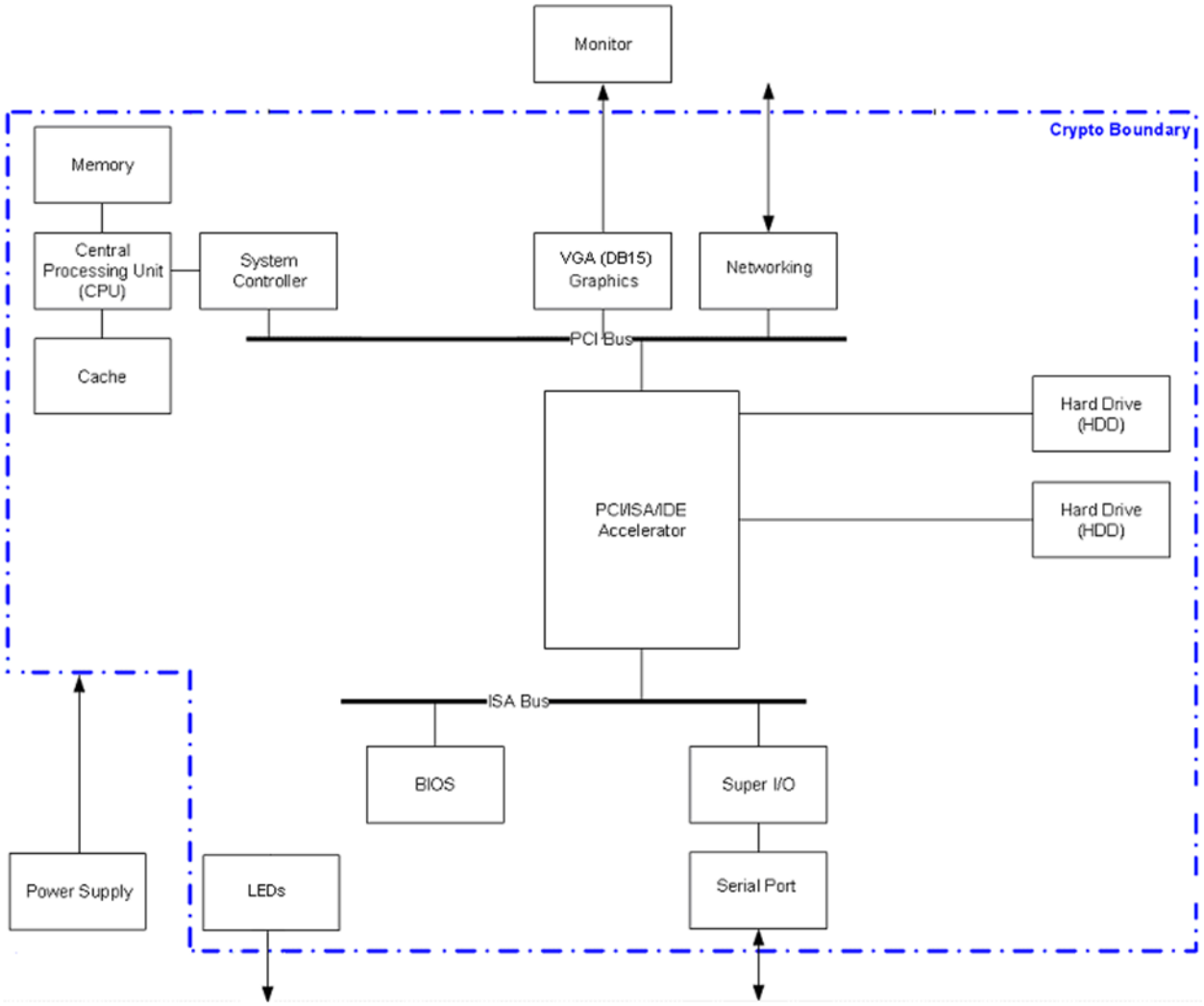


**Figure 2 – Block Diagram of ESKM**

### 2.2.1 FIPS Mode of Operation

In the FIPS mode of operation, the module implements the following Approved algorithms:

- Advanced Encryption Standard (AES) encryption and decryption: 128, 192, and 256 bits, in Electronic Codebook (ECB), Cipher Block Chaining (CBC), Counter (CTR), Key Wrap (KW) modes, CMAC generation and verification, 128 and 256 bits Galois/Counter Mode (GCM)[1] [2] encryption and decryption, and 256 bits in Counter with CBC-MAC (CCM) (Certificate #3995)
- Triple Data Encryption Standard (Triple-DES) encryption and decryption: 3-key, in ECB and CBC modes, and CMAC generation and verification, (Certificate #2194)
- Secure Hash Algorithm (SHA)-1, SHA-224, SHA-256, SHA-384, SHA-512 (Certificate #3297)
- Keyed-Hash Message Authentication Code (HMAC)-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 (Certificate #2609)
- Rivest, Shamir, and Adleman (RSA) FIPS 186-4 key generation,PKCS#1 v1.5 signature generation, and signature verification: 2048 and 3072 bits (Certificate #2051)
- RSA Decryption Primitive (RSADP) (CVL Certificate #821)
- TLS Key Derivation Function (KDF) (CVL Certificate #820)
- SSH KDF (CVL Certificates #822)
- Deterministic Random Bit Generator (DRBG) using AES in CTR mode for KMS (Certificate #1186)
- Deterministic Random Bit Generator (DRBG) using AES in CTR mode for KMIP (Certificate #1185)
- SNMP KDF (CVL Certificate # 823)
- ECDSA (Curves P256, P384) Signature Generation and Verification (Certificate #889)[3]
- ECDH primitive (Curves P256, P384) (CVL Certificate #842)
- Key Transport Scheme (AES Certificate #3995; AES-GCM, key establishment methodology provides 128 or 256 bits of encryption strength)[1], (AES Certificate #3995; AES Key Wrap, key establishment methodology provides between 128 and 256 bits of encryption strength)[4], (AES Certificate #3995 and HMAC Certificate #2609; key establishment methodology provides between 128 and 256 bits of encryption strength)[5], (Triple-DES Certificate #2194 and HMAC Certificate #2609; key establishment methodology provides 112 bits of encryption strength)[5].

In the FIPS mode of operation, the module implements the following non-Approved but allowed algorithms and protocols:

- A non-Approved Non-Deterministic Random Number Generator (NDRNG) to seed the DRBG.[6]
- The following commercially-available protocols for key establishment. The protocol algorithms have been tested by the CAVP (see certificate #s above) but the protocol implementations themselves have not been reviewed or tested by the CAVP or CMVP.
    - Transport Layer Security (TLS) protocol using RSA 2048 bits for key transport (key wrapping: key establishment methodology provides 112 bits of encryption strength), or using EC Diffie-Hellman for key agreement (key agreement; key establishment methodology provides 128 or 192 bits of encryption strength).[7]

---

[1] AES GCM is only used as part of TLS 1.2 cipher suites conformant to IG A.5, RFC 5288 and SP 800-52 which are listed in Table 14 of this document.

[2] If the module's power is lost and then restored, new GCM keys will be negotiated (to meet IG A.5).

[3] The module's cryptographic library provides support for ECDSA Public Key Generation, but this functionality is not utilized in this version of the module.

[4] KMIP clients can elect to use AES-KW to encrypt the key block.

[5] Authenticated and encrypted key transport as part of SSH and TLS protocols.

[6] The module generates a minimum of 256 bits of entropy before generating keys.

[7] No parts of this protocol other than the KDF have been reviewed or tested by the CAVP.

- SSHv2 protocol using Diffie-Hellman key agreement (the Diffie-Hellman key establishment scheme provides 112 bits of security) or ECDH key agreement (the ECDH key establishment scheme provides 128 or 192 bits of security).[7]

### 2.2.2 Non-FIPS Mode of Operation

In the non-FIPS mode of operation, the module also implements the following non-Approved algorithms:

- DES
- MD5
- RC4
- RSA providing less than 112bits of security strength for signature generation and verification, and key establishment as well as the above listed protocols for key establishment.

## 2.3 Module Interfaces

FIPS 140-2 defines four logical interfaces:

- Data Input
- Data Output
- Control Input
- Status Output

The module features the following physical ports and LEDs:

- Serial port (RS232 DB9)
- Ethernet 10/100/1000 RJ-45 ports (Network Interface Card [NIC], quantity: 4)
- Monitor port (VGA DB15)
- Power input (100-240VAC)
- LEDs (four on the front panel and three on the rear panel)

The logical interfaces and their physical port mappings are described in Table 2.

**Table 2 – Logical Interface and Physical Ports Mapping**

| Logical Interface | Physical Ports |
|---|---|
| Data Input | Serial, Ethernet |
| Data Output | Monitor, serial, Ethernet |
| Control Input | Serial, Ethernet |
| Status Output | Monitor, serial, Ethernet, LEDs |

There are no ports on the front panel. There are four LEDs on the front panel. See Figure 3.


**Figure 3 – Front Panel LEDs**

Descriptions of the LEDs are given in Table 3.

**Table 3 – Front Panel LED Definitions**

| Item | Description | Status |
|---|---|---|
| 1 | Unit Identifier (UID) LED/button | Blue = Identification is activated.<br>Off = Identification is deactivated. |
| 2 | Power/Standby LED | Green = System is on.<br>Amber = System is in standby, but power is still applied.<br>Off = Power cord is not attached, power supply failure has occurred, no power supplies are installed, facility power is not available. |
| 3 | Aggregate Network LED | Solid green = Link to network<br>Flashing green = Network activity<br>Off = No network connection |
| 4 | System Health LED | Green = System health is normal.<br>Amber = System health is degraded. Red = System health is critical.<br>Off = System health is normal (when in standby mode). |

The components on the rear panel are illustrated in

Figure 4.

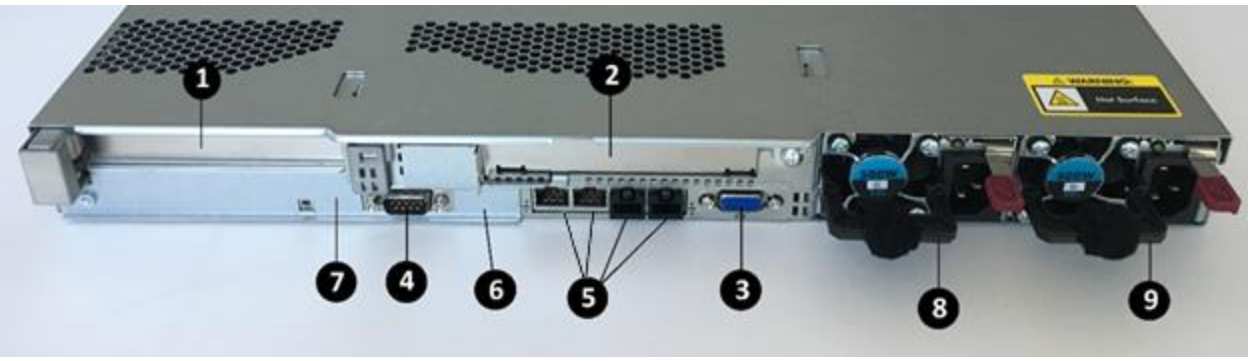


**Figure 4 – Rear Panel Components**

Descriptions of components on the rear panel are given in Table 4.

**Table 4 – Rear Panel Components Descriptions**

| Item | Definition |
|---|---|
| 1 | Slot 1 PCIe 3.0 x 16 (Blocked) |
| 2 | Slot 2 PCIe 3.0 x 16 (Blocked) |
| 3 | Video connector |
| 4 | Serial connector |
| 5 | NIC 4 connector (Not used)[8]<br>NIC 3 connector (Not used)[8]<br>NIC 2 connector<br>NIC 1 connector |
| 6 | iLO connector (Blocked) |
| 7 | USB connectors (4) (Blocked) |
| 8 | Power supply bay 1 |
| 9 | Power supply bay 2 |

The three LEDs on the rear panel are illustrated in Figure 5.

[8] NIC 3 and NIC 4 are not used by the module. RJ-45 plugs are installed as a reminder to the operator.

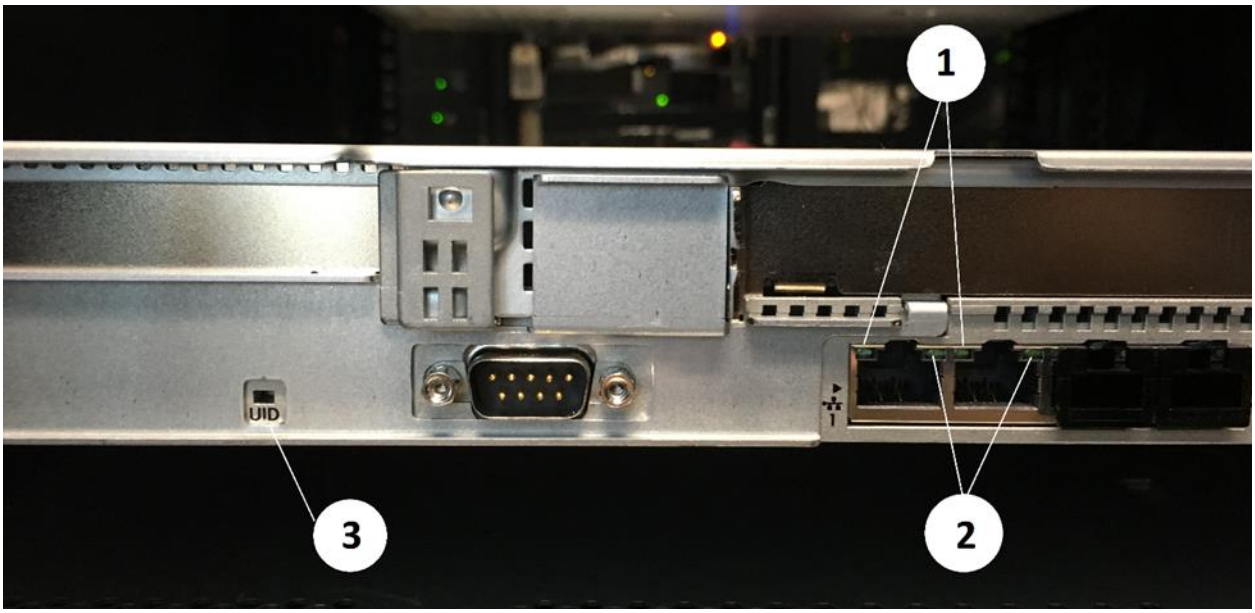


**Figure 5 – Rear Panel LEDs**

Descriptions of LEDs on the rear panel are given in Table 5.

**Table 5 – Rear Panel LED Definitions**

| Item | Description | Status |
|---|---|---|
| 1 | Standard NIC activity LED for NIC 1 and NIC 2 | Green = Activity exists.<br>Flashing green = Activity exists.<br>Off = No activity exists. |
| 2 | Standard NIC link LED for NIC 1 and NIC 2 | Green = Link exists.<br>Off = No link exists. |
| 3 | UID LED | Solid blue = Identification is activated.<br>Off = Identification is deactivated. |

## 2.4 Roles, Services, and Authentication

The module supports four authorized roles:

- Crypto-Officer
- User
- Micro Focus User
- Cluster Member

All roles require identity-based authentication.

### 2.4.1 Crypto-Officer Role

The Crypto-Officer accesses the module via the Web Management Console and/or the Command Line Interface (CLI). This role provides all services that are necessary for the secure management of the module. Table 6 shows the services for the Crypto-Officer role under the FIPS mode of operation. The

purpose of each service is shown in the first column (“Service”), and the corresponding function is described in the second column (“Description”). The Critical Security Parameters (CSPs) in the rightmost column correspond to the keys and other CSPs introduced in Section 2.7.1.

**Table 6 – Crypto-Officer Services**

| Service | Description | Keys/CSPs |
|---|---|---|
| Authenticate to ESKM | Authenticate to ESKM with a username and the associated password and/or certificate/public key | Crypto-Officer passwords – read;<br>SSH keys – read, write (only DH, ECDH, and session keys).<br>TLS keys – read, write (only MS, ECDH, and session keys). |
| Perform first-time initialization | Configure the module when it is used for the first time | Crypto-Officer (admin) password – write;<br>Krsa private – write;<br>Log signing keys – write;<br>KRsaPub – write;<br>KRsaPriv – write. |
| Configure FIPS mode | Enable/disable FIPS mode | None |
| Manage CSPs | Manage all client CSPs that are stored within the module. This includes the generation, storage, export (only public keys), import, and zeroization of keys. | Client CSPs – write, read, delete;<br>PKEK – write, read, delete. |
| Manage clusters | Manage all clusters that are defined within the module. This includes the creation, joining, and removal of a cluster from the module. | Cluster Member passwords –write, delete<br>Cluster key –write, read, delete |
| Manage services | Manage all services supported by the module. This includes the starting and stopping of all services. | SNMPv3 password – write, delete |
| Manage operators | Create, modify, or delete module operators (Crypto-Officers and Users). | Crypto-Officer passwords – write, delete; User passwords – write, delete |
| Manage certificates | Create/import/delete certificates | KRsaPub – write, read, delete;<br>KRsaPriv – write, read, delete;<br>CARsaPub – write, read, delete;<br>CARsaPriv – write, read, delete;<br>Client RSA public keys – read;<br>KECDSAPub – write, read, delete;<br>KECDSAPriv – write, read, delete;<br>CAECDSAPub – write, read, delete;<br>Client ECDSA public keys – read. |
| Reset factory settings | Rollback to the default firmware shipped with the module | All CSPs – delete |
| Restore default configuration | Delete the current configuration file and restores the default configuration settings | None |
| Restore configuration file | Restore a previously backed up configuration file | None |

| Service | Description | Keys/CSPs |
|---|---|---|
| Backup configuration file | Back up a configuration file | None |
| Sign logs | Sign module's logs using log signing key. | Log signing keys – read |
| Export logs | Export module's logs. | None |
| Zeroize all keys/CSPs | Zeroize all keys and CSPs in the module | All keys and CSPs – delete |
| Load firmware[9] | Load firmware onto the module | Firmware signature key – read |

[9] New firmware versions within the scope of this validation must be validated through the FIPS 140-2 CMVP. Any other firmware loaded into this module is out of the scope of this validation and requires a separate FIPS 140-2 validation.

### 2.4.2 User Role

The User role is associated with external applications or clients that connect to the KMS via its XML interface or to the KMIP interface. Users in this role may exercise services—such as key generation and management—based on configured or predefined permissions. See Table 7 for details. The keys and CSPs in the rightmost column correspond to the keys and CSPs introduced in Section 2.7.1.

**Table 7 – User Services**

| Service | Description | Keys/CSPs |
|---|---|---|
| Authenticate to ESKM | Authenticate to ESKM with credentials such as username and password (in addition to the certificate during TLS) | User passwords – read.<br>TLS keys – read, write (only MS, ECDH, and session keys). |
| Generate key | Generate a cryptographic key | Client keys – write;<br>PKEK – write. |
| Modify CSP attributes | Update/add/delete attributes | None |
| Delete CSP | Delete a CSP | Client CSP – delete;<br>PKEK – delete. |
| Query CSP attributes | Query a CSP's attributes that the User is allowed to access | None |
| Query | Query the module's supported capabilities | None |
| Import CSP | Import CSPs such as keys, secret data | Client CSP – write;<br>PKEK – write. |
| Export CSP | Export a CSP, such as a cryptographic key, certificate, and other KMIP objects | Client CSP – read;<br>PKEK – read. |
| Get certificate info | Return a list of local CAs including the certificate status, certify and re-certify | None |
| Clone key | Clone an existing key under a different key name | Client CSP – write, read;<br>PKEK – write, read. |
| Generate random number | Generate a random number | DRBG seed – write, read, delete<br>DRBG v – write, read, delete<br>DRBG entropy input – write, read, delete<br>DRBG key – write, read, delete |
| Crypto operation | Perform a cryptographic operation using the client key | Client key – write;<br>PKEK – read. |

| Service | Description | Keys/CSPs |
|---|---|---|
| Re-key | Create a new version of the client key | Client key – write;<br>PKEK – read. |
| Activate CSP | Activate CSP | None |
| Revoke CSP | Revoke CSP | None |

### 2.4.3 Micro Focus User Role

The Micro Focus User role can reset the module to an uninitialized state in the event that all Crypto-Officer passwords are lost, or when a self-test permanently fails. See Table 8. The keys and CSPs in the rightmost column correspond to the keys and CSPs introduced in Section 2.7.1.

**Table 8 – Micro Focus User Services**

| Service | Description | Keys/CSPs |
|---|---|---|
| Authenticate to the module | Authenticate to ESKM with a signed token | Micro Focus User RSA public key – read |
| Reset factory settings | Rollback to the default firmware shipped with the module | All keys/CSPs – delete |
| Restore default configuration | Delete the current configuration file and restores the default configuration settings | None |
| Zeroize all keys/CSPs | Zeroize all keys/CSPs in the module | All keys/CSPs – delete |

### 2.4.4 Cluster Member Role

The Cluster Member role is associated with other ESKMs that can connect to this ESKM and access cluster services. See Table 9. The keys and CSPs in the rightmost column correspond to the keys and CSPs introduced in Section 2.7.1.

**Table 9 – Cluster Member Services**

| Service | Description | Keys/CSPs |
|---|---|---|
| Authenticate Cluster Member | Authenticate to ESKM via TLS | Cluster Member passwords – read; Cluster key – read; Cluster Member RsaPub – read |
| Receive Configuration File | Update the module's configuration settings | None |
| Zeroize Key | Delete a specific key | Cluster key – delete |
| Backup Configuration File | Back up a configuration file | None |

### 2.4.5 Authentication

The module performs identity-based authentication for the four roles. Three authentication schemes are used: authentication with certificate in TLS, public key authentication via SSH, and authentication with password. See Table 10 for a detailed description.

**Table 10 – Roles and Authentications**

| Role | Authentication |
|---|---|
| Crypto-Officer | Username and password with optional digital certificate or public key |
| User | Username and password and/or digital certificate |
| Micro Focus User | Digital certificate |
| Cluster Member | Authenticates with Cluster Password and Cluster Key (certificates) |

Table 11 provides a mapping of the security strength for each authentication scheme.

**Table 11 – Strength of Authentication Mechanisms**

| Scheme | Strength |
|---|---|
| Digital Certificate and Public Key | A 2048 bit RSA key provides 112 bits of security. The probability of a successful random guess is $2^{-112}$ which is significantly less than 1 in 1,000,000 ($10^{-6}$).<br>The module's network interface supports a maximum theoretical bandwidth of $10^{10}$ bits per second. Assuming a 2048-bit key size, at least 2048 bits of data must be transmitted for one authentication attempt.[10] Therefore, in a worst case scenario the maximum number of attempts possible in a minute is 60 x ($10^{10}$ / 2048) = 292968750. The probability of a successful attempt in one minute is 292968750 x $2^{-112}$ ≈ 5.64237x$10^{-26}$ which is significantly less than 1 in 100,000.<br><br>An ECDSA key using NIST Curve P256 provides equivalent to 128 bits of security. The probability of a successful random guess is $2^{-128}$ which is significantly less than 1 in 1,000,000 ($10^{-6}$).<br>The module's network interface supports a maximum theoretical bandwidth of $10^{10}$ bits per second. Assuming a 256-bit key size, at least 256 bits of data must be transmitted for one authentication attempt.[10] Therefore, in a worst case scenario the maximum number of attempts possible in a minute is 60 x ($10^{10}$ / 256) = 2343750000. The probability of a successful attempt in one minute is 2343750000 x $2^{-128}$ ≈ 6.88766x$10^{-30}$ which is significantly less than 1 in 100,000 ($10^{-5}$).<br><br>An ECDSA key using NIST Curve P384 provides equivalent to 192 bits of security. The probability of a successful random guess is $2^{-192}$ which is significantly less than 1 in 1,000,000.<br>The module's network interface supports a maximum theoretical bandwidth of $10^{10}$ bits per second. Assuming a 384-bit key size, at least 384 bits of data must be transmitted for one authentication attempt.[10] Therefore, in a worst case scenario the maximum number of attempts possible in a minute is 60 x ($10^{10}$ / 384) = 1562500000. The probability of a successful attempt in one minute is 1562500000 x $2^{-192}$ ≈ 2.48921x$10^{-49}$ which is significantly less than 1 in 100,000 ($10^{-5}$). |

[10] In reality, much more data is required to be transmitted for each authentication attempt.

| Scheme | Strength |
|---|---|
| Password | Passwords in the module must consist of eight or more characters from the set of 90 human-readable numeric, alphabetic (upper and lower case), and special character symbols. In a worst case scenario (password consisting of all numbers) the size of the password space is $10^8$. The probability of a successful random guess is $10^{-8}$ which is less than 1 in 1,000,000 ($10^{-6}$).<br>After five unsuccessful password attempts, the module is locked down for 60 seconds. The probability of a successful password attempt in one minute is 5 x $10^{-8}$ which is less than 1 in 100,000 ($10^{-5}$). |

### 2.4.6 Unauthenticated Services

The following services do not require authentication:

- SNMP statistics
- FIPS status services
- Health check services
- Network Time Protocol (NTP) services
- Initiation of self-tests by rebooting the ESKM
- Negotiation of the XML protocol version for communications with the KMS

SNMP is used only for sending statistical information (SNMP traps). FIPS status and health check are status-report services, unrelated to security or cryptography. NTP is a date/time synchronization service that does not involve keys or CSPs. Initiation of self-tests and negotiation of the XML protocol version do not involve keys or CSPs.

The services listed above for each role comprise the entire set of services available in non-FIPS mode.

### 2.4.7 Non-approved Services

The following services are available in non-FIPS mode:

- File Transfer Protocol (FTP) for importing certificates and downloading and restoring backup files
- Lightweight Directory Access Protocol (LDAP) authentication
- Use of the following algorithms with KMS or KMIP servers: RC4, MD5, DES, RSA-512, RSA-768, RSA-1024
- SNMPv1/SNMPv2
- SSL 3.0
- RSA encryption and decryption operations (note, however, that RSA encryption and decryption associated with TLS handshakes and Sign and Sign Verify *are* allowed in FIPS Mode)

## 2.5 Physical Security

The module was tested and found conformant to the EMI/EMC requirements specified by Title 47 of the Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class A (that is, for business use).

The Enterprise Secure Key Manager is a multi-chip standalone cryptographic module. The entire contents of the module, including all hardware, software, firmware, and data, are enclosed in a metal case. The case is opaque and must be sealed using a tamper-evident label in order to prevent the case cover from

being removed without signs of tampering. Two pick-resistant locks are installed on the module's front bezel to protect the front interfaces, including the power switch, from unauthorized access. All circuits in the module are coated with commercial standard passivation. Once the bezel is locked and the module has been configured to meet FIPS 140-2 Level 2 requirements, the module cannot be accessed without signs of tampering. See Section 3.3 – Physical Security Assurance of this document for more information.

## 2.6 Operational Environment

The operational environment requirements do not apply to the Enterprise Secure Key Manager—the module does not provide a general purpose operating system.

## 2.7 Cryptographic Key Management

### 2.7.1 Keys and CSPs

The SSH and TLS protocols employed by the FIPS mode of the module are security-related. Table 12 and Table 13 introduce cryptographic keys, key components, and CSPs involved in the two protocols, respectively.

**Table 12 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs for SSH**

| Key | Key Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| DH public param | 2048-bit Diffie-Hellman public parameters | Generated by DRBG during session initialization | In plaintext | In volatile memory | Upon session termination | Negotiate SSH Ks and SSH Khmac |
| DH private param | 2048-bit Diffie-Hellman private parameters | Generated by DRBG during session initialization | Never | In volatile memory | Upon session termination | Negotiate SSH Ks and SSH Khmac |
| ECDH public param | P256 or P384 EC Diffie-Hellman public parameters | Generated by DRBG during first-time initialization | In plaintext | In volatile memory | Upon session termination | Negotiate SSH Ks and SSH Khmac |
| ECDH private param | P256 or P384 EC Diffie-Hellman private parameters | Generated by DRBG during first-time initialization | Never | In volatile memory | Upon session termination | Negotiate SSH Ks and SSH Khmac |
| Krsa public | 2048-bit RSA public keys | Generated by DRBG during first-time initialization | In plaintext | In non-volatile memory | At operator delete or zeroize request | Verify the signature of the server's message. |
| Krsa private | 2048-bit RSA private keys | Generated by DRBG during first-time initialization | Never | In non-volatile memory | At operator delete or zeroize request | Sign the server's message. |
| Krsa public auth | 2048-, 3072-bit RSA public key | Imported by the Crypto-Officer | In plaintext | In non-volatile memory | At operator delete or zeroize request | Authenticate Crypto-Officer |
| SSH Ks | SSH session 3-key Triple-DES key, 128-, 192-, 256-bit AES key | Diffie-Hellman key agreement | Never | In volatile memory | Upon session termination or when a new Ks is generated (after a certain timeout) | Encrypt and decrypt data |

| Key | Key Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| SSH Khmac | SSH session 512-bit HMAC key | Diffie-Hellman key agreement | Never | In volatile memory | Upon session termination or when a new Khmac is generated (after a certain timeout) | Authenticate data |

**Table 13 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs for TLS**

| Key | Key Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| ECDH public param | P256 or P384 EC Diffie-Hellman public parameters | Generated by DRBG during session initialization | In plaintext | In volatile memory | Upon session termination | Establish TLS Pre-MS |
| ECDH private param | P256 or P384 EC Diffie-Hellman private parameters | Generated by DRBG during session initialization | Never | In volatile memory | Upon session termination | Establish TLS Pre-MS |
| Pre-MS | TLS pre-master secret | Input in encrypted form from TLS client | Never | In volatile memory | Upon session termination | Derive MS |
| MS | TLS master secret | Derived from Pre-MS using FIPS Approved key derivation function | Never | In volatile memory | Upon session termination | Derive TLS Ks and TLS Khmac |
| KRsaPub | Server RSA public key (2048-bit) | Generated by DRBG during first-time initialization | In plaintext a X509 certificate | In non-volatile memory | At operator delete request | Client encrypts Pre-MS. Client verifies server signatures |
| KECDSAPub | Server ECDSA public key (P-256, P-384) | Externally generated; imported by Crypto Officer over TLS | In plaintext a X509 certificate | In non-volatile memory | At operator delete request | Client encrypts Pre-MS. Client verifies server signatures |
| KRsaPriv | Server RSA private key (2048-bit) | Generated by DRBG during first-time initialization | Never | In non-volatile memory | At operator delete or zeroize request | Server decrypts Pre-MS. Server generates signatures |
| KECDSAPriv | Server ECDSA private key (P-256, P-384) | Externally generated; imported by Crypto Officer over TLS | Never | In non-volatile memory | At operator delete or zeroize request | Server decrypts Pre-MS. Server generates signatures |
| CARsaPub | Certificate Authority (CA) RSA public key ( 2048-bit) | Generated by DRBG during first-time initialization | In plaintext | In non-volatile memory | At operator delete request | Verify CA signatures |

| Key | Key Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| CAECDSAPub | Certificate Authority (CA) ECDSA public key (P-256, P-384) | Externally generated; imported by Crypto Officer over TLS | In plaintext | In non-volatile memory | At operator delete request | Verify CA signatures |
| CARsaPriv | CA RSA private key ( 2048-bit) | Generated by DRBG during first-time initialization | Never | In non-volatile memory | At operator delete or zeroize request | Sign server certificates |
| Cluster Member RsaPub | Cluster Member RSA public key ( 2048-bit) | Input in plaintext | Never | In volatile memory | Upon session termination | Verify Cluster Member signatures |
| TLS Ks | TLS session 128-bit AES, 256-bit AES, or 192-bit Triple-DES symmetric key(s) | Derived from MS | Never | In volatile memory | Upon session termination | Encrypt and decrypt data travelling within TLS tunnel. |
| TLS Khmac | TLS session HMAC-SHA1, HMAC-SHA256, or HMAC-SHA384 key(s) | Derived from MS | Never | In volatile memory | Upon session termination | Authenticate data travelling within TLS tunnel. |

Table 14 details all cipher suites supported by the TLS protocol implemented by the module. The suite names in the first column match the definitions in RFC 2246 and RFC 4346.

**Table 14 – Cipher Suites Supported by the Module's TLS Implementation in FIPS Mode**

| Suite Name | Authentication | Key Transport | Symmetric Cryptography | Hash |
|---|---|---|---|---|
| TLS_RSA_WITH_AES_256_CBC_SHA | RSA | RSA | AES (256-bit) | SHA-1 |
| TLS_RSA_WITH_AES_128_CBC_SHA | RSA | RSA | AES (128-bit) | SHA-1 |
| TLS_RSA_WITH_TDES_EDE_CBC_SHA | RSA | RSA | Triple-DES (3-key) | SHA-1 |
| TLS_RSA_WITH_AES_128_CBC_SHA256 | RSA | RSA | AES (128-bit) | SHA-256 |
| TLS_RSA_WITH_AES_256_CBC_SHA256 | RSA | RSA | AES (256-bit) | SHA-256 |
| TLS_RSA_WITH_AES_128_GCM_SHA256 | RSA | RSA | AES (128-bit) | SHA-256 |
| TLS_RSA_WITH_AES_256_GCM_SHA384 | RSA | RSA | AES (256-bit) | SHA-384 |
| TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 | ECDSA | ECDHE | AES (128-bit) | SHA-256 |
| TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 | ECDSA | ECDHE | AES (256-bit) | SHA-384 |

| Suite Name | Authentication | Key Transport | Symmetric Cryptography | Hash |
|---|---|---|---|---|
| TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 | RSA | ECDHE | AES (256-bit) | SHA-384 |

Other CSPs are tabulated in Table 15.

**Table 15 – Other Cryptographic Keys, Cryptographic Key Components, and CSPs**

| Key | Key Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| Client AES key | 128, 192 or 256-bit AES key | Generated by DRBG | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | Encrypted with PKEK in non-volatile memory | Per client's request or zeroize request | Encrypt plaintexts/decrypt ciphertexts |
| Client AES CMAC key | 128, 192 or 256-bit AES CMAC key | Generated by DRBG | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | Encrypted with PKEK in non-volatile memory | Per client's request or zeroize request | Encrypt plaintexts/decrypt ciphertexts |
| Client TDES key | Triple-DES key | Generated by DRBG | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | Encrypted with PKEK in non-volatile memory | Per client's request or zeroize request | Encrypt plaintexts/decrypt ciphertexts |
| Client TDES CMAC key | Triple-DES CMAC key | Generated by DRBG | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | Encrypted with PKEK in non-volatile memory | Per client's request or zeroize request | Encrypt plaintexts/decrypt ciphertexts |
| Client RSA public keys | RSA public key | Generated by DRBG | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | Encrypted with PKEK in non-volatile memory | At operator delete | Verify signatures |
| Client ECDSA public keys | ECSDA public key | Input in ciphertext over TLS | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | Encrypted with PKEK in non-volatile memory | At operator delete | Verify signatures |
| Client RSA keys | RSA private keys | Generated by DRBG | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | Encrypted with PKEK in non-volatile memory | Per client's request or zeroize request | Sign messages |

| Key | Key Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| Client ECDSA keys | ECDSA private key | Input in ciphertext over TLS | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | Encrypted with PKEK in non-volatile memory | Per client's request or zeroize request | Sign messages |
| Client HMAC keys | HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, or HMAC-SHA512 keys | Generated by DRBG | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | Encrypted with PKEK in non-volatile memory | Per client's request or zeroize request | Compute keyed-MACs |
| Client certificate | X.509 certificate | Input in ciphertext over TLS | Via TLS in encrypted form (encrypted with TLS Ks) per client's request | In non-volatile memory | Per client's request or by zeroize request | Encrypt data/verify signatures |
| Crypto-Officer passwords | Character string | Input in ciphertext over TLS | Never | In non-volatile memory | At operator delete or by zeroize request | Authenticate Crypto-Officer |
| User passwords | Character string | Input in ciphertext over TLS | Never | In non-volatile memory | At operator delete or by zeroize request | Authenticate User |
| Cluster Member password | Character string | Input in ciphertext over TLS | Never | In non-volatile memory | At operator delete or zeroize request | When a device attempts to become a Cluster Member |
| Micro Focus User RSA public key | 2048-bit RSA public key | Input in plaintext at factory | Never | In non-volatile memory | At installation of a patch or new firmware | Authenticate Micro Focus User |
| Cluster key | Character string | Input in ciphertext over TLS | Via TLS in encrypted form | In non-volatile memory | At operator delete or by zeroize request | Authenticate Cluster Member |
| Log signing keys | 2048-bit RSA public and private keys | Generated by DRBG at first-time initialization | Never | In non-volatile memory | When new log signing keys are generated on demand by Crypto-Officer | Sign logs and verify signature on logs |
| DRBG entropy input | PRNG input | Generated by non-Approved RNG or input in ciphertext over TLS | Never | In volatile memory | When module is powered off | Initialize DRBG |

| Key | Key Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| DRBG seed | DRBG seed | Generated as part of derivation function for SP 800-90A CTR_DRBG | Never | In volatile memory | When module is powered off | Initialize DRBG |
| DRBG v | DRBG Internal State | Generated as part of derivation function for SP 800-90A CTR_DRBG | Never | In volatile memory | When module is powered off | DRBG Internal State |
| DRBG key | DRBG Internal State | AES key used for SP 800-90A CTR_DRBG | Never | In volatile memory | When module is powered off | DRBG Internal State |
| PKEK | 256-bit AES key | Generated by DRBG | In encrypted form for backup purposes only | In non-volatile memory | At operator delete or by zeroize request | Encrypt Client CSP |
| SNMPv3 password | Shared secret, at least 8 characters | Input in ciphertext over TLS | Never | In non-volatile memory | At operator delete or by zeroize request | SNMPv3 authentication |
| SNMPv3 session key | 128-bit AES | Derived shared secret | Never | In volatile memory | When the module is powered off | Encrypt/decrypt SNMP traffic |
| Firmware signature key | 2048-bit RSA public key | Input in plaintext at factory | Never | In non-volatile memory | At installation of a patch or new firmware | Verify firmware signature |

### 2.7.2 Key Generation

The module uses the DRBG (AES in CTR mode) as specified in SP 800-90A to generate cryptographic keys. This DRBG is a FIPS 140-2 Approved RNG as specified in Annex C to FIPS 140-2.

### 2.7.3 Key/CSP Zeroization

All ephemeral keys are stored in volatile memory in plaintext. Ephemeral keys are zeroized when they are no longer used. Other keys and CSPs are stored in non-volatile memory with client CSPs being stored in encrypted form.

To zeroize all keys and CSPs in the module, the Crypto-Officer should execute the reset factory settings zeroize command at the serial console interface. For security reasons, this command is available only through the serial console.

Since the zeroization process can take just over one minute, the Crypto-Officer must remain with the physical module until the zeroization operation is complete.

## 2.8 Self-Tests

The device implements two types of self-tests: power-up self-tests and conditional self-tests.

Power-up self-tests include the following tests:

- Firmware integrity tests (RSA 2048-bit signature verification)
- Known Answer Test (KAT) on Triple-DES (encrypt and decrypt, ECB mode, 3-Key)
- KAT on AES (encrypt and decrypt, ECB mode, 128-bit key; this covers the KAT requirement for all AES modes although the GCM and Key Wrap modes are additionally tested)
- KAT on AES GCM (encrypt and decrypt, 256-bit key)
- KAT on AES Key Wrap (authenticated encryption and authenticated decryption, 128-, 192-, 256-bit key)
- KAT on HMAC (one KAT per SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512)
- KAT on SHA covered by above HMAC KATs per IG 9.1
- KAT on DRBG for KMS (CTR_DRBG, 256-bit AES with derivation function)
- KAT on DRBG for KMIP (CTR_DRBG, 256-bit AES with derivation function)
- KAT on Diffie-Hellman (2048-bit prime modules with 256-bit prime subgroup, shared secret calculation)
- KAT on SSH Key Derivation Function (2048-bit shared secret)
- KAT on TLS Key Derivation Function (TLS 1.0 with SHA-1, TLS 1.1 with SHA-256, TLS 1.1/1.2 with SHA-384)
- KAT on RSA signature generation and verification (sign, verify, encrypt, decrypt using 2048-bit key , SHA-256)
- KAT on RSA Decryption Primitive (decrypt, 2048-bit)
- KAT on SNMP Key Derivation Function
- KAT on ECDSA sign/verify
- KAT on ECDH

Conditional self-tests include the following tests:

- Pairwise consistency test for new RSA keys
- Continuous random number generator test on DRBG (for both KMS and KMIP)
- Health Checks per SP 800-90A Section 11.3 (for both KMS and KMIP)
- Continuous random number generator test on non-Approved RNG
- Diffie-Hellman pairwise consistency test
- Diffie-Hellman primitive test
- ECDH pairwise consistency test
- ECDH primitive test
- Firmware load test

The module has two error states: a Soft Error state and a Fatal Error state. When one or more power-up self-tests fail, the module enters the Fatal Error state. When a conditional self-test fails, the module enters the Soft Error state. See Section 3 of this document for more information.

## 2.9 Mitigation of Other Attacks

This section is not applicable. No claim is made that the module mitigates against any attacks beyond the FIPS 140-2 Level 2 requirements for this validation.

# 3 Secure Operation

The Enterprise Secure Key Manager meets Level 2 requirements for FIPS 140-2. The sections below describe how to place and keep the module in the FIPS mode of operation.

## 3.1 Initial Setup

The device should be unpacked and inspected according to the *Installation Guide*. The *Installation Guide* also contains installation and configuration instructions, maintenance information, safety tips, and other information.

## 3.2 Initialization and Configuration

### 3.2.1 First-Time Initialization

When the module is turned on for the first time, it will prompt the operator for a password for a default Crypto-Officer. The module cannot proceed to the next state until the operator provides a password that conforms to the password policy described in Section 2.4.5. The default username associated with the entered password is “admin”.

During the first-time initialization, the operator must configure minimum settings for the module to operate correctly. The operator will be prompted to configure the following settings via the serial interface:

- Date, Time, Time zone
- IP Address/Netmask
- Hostname
- Gateway
- Management Port

### 3.2.2 FIPS Mode Configuration

In order to comply with FIPS 140-2 Level 2 requirements, the following functionality must be disabled on the ESKM:

- Global keys
- File Transfer Protocol (FTP) for importing certificates and downloading and restoring backup files
- Lightweight Directory Access Protocol (LDAP) authentication
- Use of the following algorithms: RC4, MD5, DES, RSA-512, RSA-768, RSA-1024
- SSL 3.0
- RSA encryption and decryption operations (note, however, that RSA encryption and decryption associated with TLS handshakes and Sign and Sign Verify *are* permitted)

These functions need not be disabled individually. There are two approaches to configuring the module such that it works in the Approved FIPS mode of operation:

Through a command line interface, such as SSH or serial console, the Crypto-Officer should use the fips compliant command to enable the FIPS mode of operation. This will alter various server settings as described above. See Figure 6. The fips server command is used for the FIPS status server configuration. The show fips status command returns the current FIPS mode configuration.

```
labhp (config)# fips compliant
This device is now FIPS compliant.
labhp (config)# fips server
Enable FIPS Status Server [y]:
Available IP addresses:
        1. All
        2. 192.168.0.202
Local IP (1-2)[1]:
Local Port [9081]:
labhp (config)# show fips status
FIPS Compliant: Yes
```

**Figure 6 – FIPS Compliance in CLI**

In the web administration interface, the Crypto-Officer should use the "High Security Configuration" page to enable and disable FIPS compliance. To enable the Approved FIPS mode of operation, click on the "Set FIPS Compliant" button. See Figure 7. This will alter various server settings as described above.

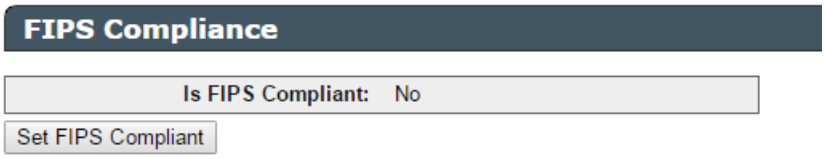


**Figure 7 – FIPS Compliance in Web Administration Interface**

In the web administration interface, the User can review the FIPS mode configuration by reading the "High Security Configuration" page.

When operating in FIPS mode, the FIPS Status Server is enabled by default and should not be disabled.

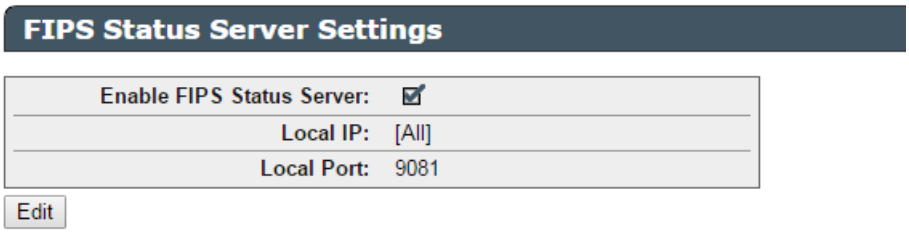


**Figure 8 – FIPS Status Server Settings in Web Administration Interface**

The Crypto-Officer must zeroize all keys when switching from the Approved FIPS mode of operation to the non-FIPS mode and vice versa.

All services are available in both the Approved FIPS mode of operation and the non-FIPS mode of operation.

## 3.3 Physical Security Assurance

One serialized tamper-evidence label has been applied during manufacturing on the metal casing. See Figure 9. The tamper-evidence label has a special adhesive backing to adhere to the module's surface and have an individual, unique serial number. It should be inspected every six months and compared to the previously-recorded serial number to verify that fresh label have not been applied to a tampered

module. If the label shows evidence of tamper, the Crypto-Officer should assume that the module has been compromised and contact Micro Focus Customer Support.

**Figure 9 – Tamper-Evidence Label on ESKM**

## 3.4 Key and CSP Zeroization

To zeroize all keys and CSPs in the module, the Crypto-Officer should execute reset factory settings zeroize command in the serial console interface. Notice that, for security reasons, the command cannot be initiated from the SSH interface.

Since the zeroization process can take just over one minute, the Crypto-Officer must remain with the physical module until the zeroization operation is complete.

When switching between different modes of operations (FIPS and non-FIPS), the Crypto-Officer must zeroize all CSPs.

## 3.5 Error State

The module has two error states: a Soft Error state and a Fatal Error state.

When a power-up self-test fails, the module will enter the Fatal Error state. When a conditional self-test fails, the module will enter the Soft Error state. The module can recover from the Fatal Error state if power is cycled or if the ESKM is rebooted. A Micro Focus User can reset the module when it is in the Fatal Error State. No other services are available in the Fatal Error state. The module can recover from the Soft Error state if power is cycled. A User can access the FIPS Status Server on port 9081 and find the error message indicating the failure of FIPS self-tests. Access to port 9081 does not require authentication.

# Acronyms

**Table 16 – Acronyms**

| Acronym | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| ANSI | American National Standard Institute |
| BIOS | Basic Input/Output System |
| CA | Certificate Authority |
| CBC | Cipher Block Chaining |
| CLI | Command Line Interface |
| CMVP | Cryptographic Module Validation Program |
| CPU | Central Processing Unit |
| CRC | Cyclic Redundancy Check |
| CSP | Critical Security Parameter |
| DES | Data Encryption Standard |
| DRBG | Deterministic Random Bit Generator |
| DSA | Digital Signature Algorithm |
| ECB | Electronic Codebook |
| ECDH | Elliptic Curve Diffie-Hellman |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| EMC | Electromagnetic Compatibility |
| EMI | Electromagnetic Interference |
| ESKM | Enterprise Secure Key Manager |
| FIPS | Federal Information Processing Standard |
| FTP | File Transfer Protocol |
| HDD | Hard Drive |
| HMAC | Keyed-Hash Message Authentication Code |
| IDE | Integrated Drive Electronics |
| iLO | Integrated Lights-Out |
| I/O | Input/Output |
| IP | Internet Protocol |
| ISA | Instruction Set Architecture |
| KAT | Known Answer Test |
| KMS | Key Management Service |
| KMIP | Key Management Interoperability Protocol |
| LDAP | Lightweight Directory Access Protocol |
| LED | Light Emitting Diode |
| MAC | Message Authentication Code |

| Acronym | Definition |
|---|---|
| N/A | Not Applicable |
| NIC | Network Interface Card |
| NIST | National Institute of Standards and Technology |
| NTP | Network Time Protocol |
| PCI | Peripheral Component Interconnect |
| RFC | Request for Comments |
| RNG | Random Number Generator |
| RSA | Rivest, Shamir, and Adleman |
| SHA | Secure Hash Algorithm |
| SNMP | Simple Network Management Protocol |
| SSH | Secure Shell |
| SSL | Secure Socket Layer |
| Triple-DES | Triple Data Encryption Standard |
| TLS | Transport Layer Security |
| UID | Unit Identifier |
| USB | Universal Serial Bus |
| VGA | Video Graphics Array |
| XML | Extensible Markup Language |