Ribbon Communications, Inc.

SBC SWe Session Border Controller

Version: 10.01.06

FIPS 140-3 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.6

Prepared for:

Ribbon Communications, Inc.
4 Technology Park Drive
Westford, MA 01886
United States of America

Phone: +1 855 467 6687
www.ribboncommunications.com

Prepared by:

Corsec Security, Inc.
12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
www.corsec.com

Table of Contents

1. General 6
1.1 Overview 6
1.2 Security Levels 6

2. Cryptographic Module Specification 8
2.1 Description 8
2.2 Tested and Vendor Affirmed Module Version and Identification 12
2.3 Excluded Components 14
2.4 Modes of Operation 14
2.5 Algorithms 14
2.6 Security Function Implementations 18
2.7 Algorithm Specific Information 23
2.8 RNG and Entropy 24
2.9 Key Generation 24
2.10 Key Establishment 25
2.11 Industry Protocols 25

3. Cryptographic Module Interfaces 26
3.1 Ports and Interfaces 26

4. Roles, Services, and Authentication 27
4.1 Authentication Methods 27
4.2 Roles 28
4.3 Approved Services 29
4.4 Non-Approved Services 39
4.5 External Software/Firmware Loaded 39

5. Software/Firmware Security 41
5.1 Integrity Techniques 41
5.2 Initiate on Demand 41

6. Operational Environment 42
6.1 Operational Environment Type and Requirements 42

7. Physical Security 43

8. Non-Invasive Security 44

9. Sensitive Security Parameters Management 45
9.1 Storage Areas 45
9.2 SSP Input-Output Methods 45
9.3 SSP Zeroization Methods 46
9.4 SSPs 46
9.5 Transitions 52

10. Self-Tests 53
10.1 Pre-Operational Self-Tests 53
10.2 Conditional Self-Tests 53

10.3 Periodic Self-Test Information 55
10.4 Error States 57
10.5 Operator Initiation of Self-Tests 57

11. Life-Cycle Assurance 58
11.1 Installation, Initialization, and Startup Procedures 58
11.2 Administrator Guidance 61
11.3 Non-Administrator Guidance 63
11.4 Design and Rules 63
11.5 End of Life 63

12. Mitigation of Other Attacks 64

Appendix A. Acronyms and Abbreviations 65

List of Tables

Table 1: Security Levels 6
Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets) 13
Table 3: Tested Operational Environments - Software, Firmware, Hybrid 13
Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid 14
Table 5: Modes List and Description 14
Table 6: Approved Algorithms - Ribbon Cryptographic Library 16
Table 7: Approved Algorithms - Ribbon SRTP KDF Library (IPP) 16
Table 8: Approved Algorithms - 16
Table 9: Approved Algorithms - Ribbon IKE KDF Library 17
Table 10: Approved Algorithms - Ribbon IKE KDF Library 17
Table 11: Approved Algorithms - Ribbon SSH KDF Library 17
Table 12: Approved Algorithms - Ribbon TLS KDF Library 17
Table 13: Approved Algorithms - Ribbon Entropy Library 17
Table 14: Vendor-Affirmed Algorithms 17
Table 15: Security Function Implementations 23
Table 16: Entropy Certificates 24
Table 17: Entropy Sources 24
Table 18: Ports and Interfaces 26
Table 19: Authentication Methods 27
Table 20: Roles 29
Table 21: Approved Services 39
Table 22: Storage Areas 45
Table 23: SSP Input-Output Methods 45
Table 24: SSP Zeroization Methods 46
Table 25: SSP Table 1 49
Table 26: SSP Table 2 52
Table 27: Pre-Operational Self-Tests 53
Table 28: Conditional Self-Tests 55
Table 29: Pre-Operational Periodic Information 56
Table 30: Conditional Periodic Information 56
Table 31: Error States 57
Table 32. Acronyms and Abbreviations 65

List of Figures

Figure 1. Typical Deployment of SBC SWe in a Network 9
Figure 2. Module Block Diagram (with Cryptographic Boundary) 11
Figure 3. Block Diagram of a GPC 12

1. General

1.1 Overview

1.1.1 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-3 cryptographic module security policy. More information is available on the module from the following sources:

- The Ribbon website (www.ribboncommunications.com) contains information on the full line of services and solutions from Ribbon.
- The search page on the CMVP website (https://csrc.nist.gov/Projects/cryptographic-module-validation-program/Validated-Modules/Search) can be used to locate and obtain vendor contact information for technical or sales-related questions about the module.

1.1.2 Document Organization

ISO/IEC 19790 Annex B uses the same section naming convention as *ISO/IEC 19790* section 7 - Security requirements. For example, Annex B section B.2.1 is named "General" and B.2.2 is named "Cryptographic module specification," which is the same as *ISO/IEC 19790* section 7.1 and section 7.2, respectively. Therefore, the format of this Security Policy is presented in the same order as indicated in Annex B, starting with "General" and ending with "Mitigation of other attacks." If sections are not applicable, they have been marked as such in this document.

1.2 Security Levels

The SBC SWe Session Border Controller is validated at the FIPS 140-3 section levels shown in the table below.

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	3
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
	Overall Level	1

Table 1: Security Levels

The module has an overall security level of 1.

2. Cryptographic Module Specification

2.1 Description

2.1.1 Purpose and Use

Ribbon Communications, Inc. (hereafter referred to as Ribbon) is a leader in IP[1] networking with proven expertise in delivering secure, reliable and scalable next-generation infrastructure and subscriber solutions. The Ribbon line of Session Border Controller (SBC) solutions help mid-sized and large enterprises take advantage of cost-saving SIP[2] trunking services by securing their network from IP-based attacks, unifying SIP-based communications and controlling traffic in the network.

The Ribbon SBC SWe Session Border Controller (SBC SWe) is a software-based, cloud-optimized SBC architected to enable and secure real-time communications in the cloud. Starting at 25 sessions and scaling to tens of thousands per instance, the unique architecture of the SBC SWe allows customers to define where on the performance curve their network needs to reside. The SBC SWe uses a "microservices" architecture designed to separate signaling, media, and transcoding to optimize virtual network resources. It also supports on-demand auto-scaling, with a feedback loop using Key Performance Indicators and the Ribbon Virtual Network Function (VNF) Manager.

The SBC SWe features the same code base, resiliency, media transcoding, and security technology found in Ribbon's hardware-based SBC 5000 Series and SBC 7000 Session Border Controllers. However, as a software solution, customers can deploy the SBC SWe as a VNF on industry-standard servers in a data center environment using a hypervisor, as a VNF in an OpenStack cloud infrastructure, or as a VNF on public cloud or hosted services.

Some of the network and security features provided by the SBC SWe are:

- Built-in media transcoding capability
- Industry-leading user interface for ease of management and ongoing operations
- Common service orchestration with Ribbon's centralized call routing and policy management for network-wide intelligence and control
- Enhanced security/encryption services to protect privacy and ensure compliance
- Load-balancing of Real Time Communications (RTC) traffic across the cloud for network efficiency
- Integrated analytics of network traffic to drive orchestration of SBC VNFs
- TLS[3], IPsec (IKEv1[4]) for signaling encryption
- Secure RTP/RTCP[5] for media encryption

[1] IP – Internet Protocol
[2] SIP – Session Initiation Protocol
[3] TLS – Transport Layer Security
[4] IKEv1 – Internet Key Exchange version 1
[5] RTCP – RTP Control Protocol

- Support for large number of protocols including IPv4, IPv6, IPv4/IPv6 interworking, SSH[6], SFTP[7], SNMP[8], HTTPS[9], RTP/RTCP, UDP[10], TCP[11], DNS[12], and ENUM[13]

Figure 1 below illustrates a typical deployment scenario of the module.

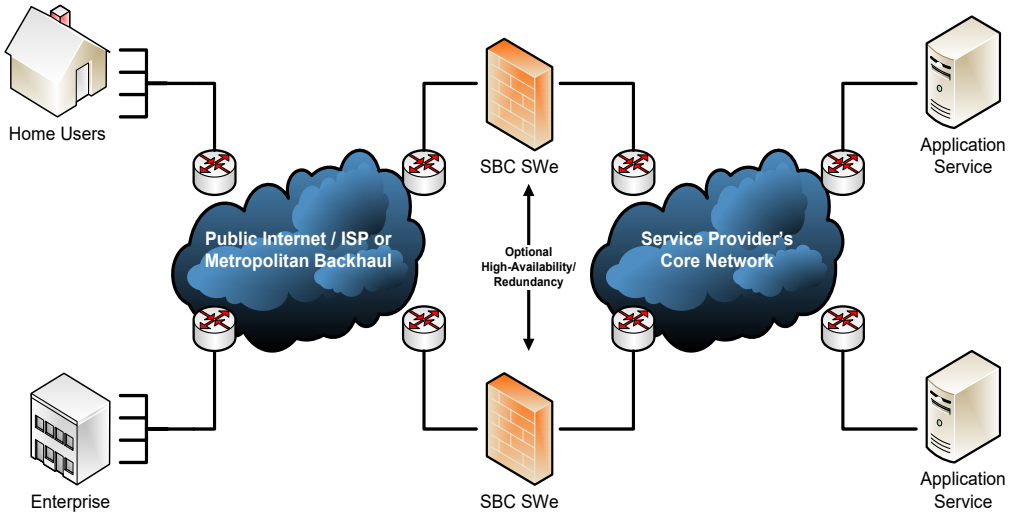

Figure 1. Typical Deployment of SBC SWe in a Network

Management of the SBC SWe is accomplished using the following tools:

- Command Line Interface (CLI), which is accessible remotely via SSH over Ethernet management ports.
- Web-based Graphical User Interface (GUI) called Embedded Management Application (EMA), which is accessible remotely via HTTPS over Ethernet management ports.

The module also provides an SNMPv3[14] interface for remote management and non-security relevant information about the module's state and statistics. In addition, the module provides an SFTP interface for transferring the Security Event log, the System Event log, release packages, tone and announcement files, CDR[15] logs, and configuration files over the virtual machine's Ethernet management ports.

These management interfaces provide authorized operators access to the module for configuration and management of all facets of the module's operation, including system configuration, troubleshooting, security, and service provisioning. Using any of the management interfaces, an operator is able to monitor, configure, control, receive report events, and retrieve logs from the SBC SWe.

[6] SSH – Secure Shell
[7] SFTP – SSH File Transport Protocol
[8] SNMP – Simple Network Management Protocol
[9] HTTPS – Hypertext Transfer Protocol Secure
[10] UDP – User Datagram Protocol
[11] TCP – Transmission Control Protocol
[12] DNS – Domain Name System
[13] ENUM – E.164 Number Mapping
[14] SNMPv3 – Simple Network Management Protocol version 3
[15] CDR – Call Detail Records

To support TLS, the module employs the following certificate management techniques:

- Local – Local certificates are credentials belonging to the SBC SWe itself, which it presents to peers in order to prove its identity. Local certificate files must be downloaded to the module before installing the certificates.

- Local-Internal – The SBC SWe generates and installs RSA key pairs and generates Certificate Signing Requests (CSR) internally. The CSR is sent to a CA, and the issued certificate is then installed on the SBC.

- Remote – Remote certificates are credentials belonging to CAs. The CA certificates contain public keys only; they do not contain the associated private keys. The CA certificates are Distinguished Encoding Rules (DER) format files.

2.1.2 Module Type

The SBC SWe Session Border Controller 10.01.06 is a Software module.

2.1.3 Module Embodiment

The SBC SWe Session Border Controller has a MultiChipStand embodiment.

2.1.4 Module Characteristics

The module does not have any additional characteristics.

2.1.5 Cryptographic Boundary

As a software appliance, the module has no physical components. Since the module has no physical characteristics, it makes use of the physical interfaces of the server hosting the virtual environment upon which the module is installed. The hypervisor controls and directs all interactions between the module and the operator, and it is responsible for mapping the module's virtual interfaces to the host server's physical interfaces.

The module's cryptographic boundary consists of all functionalities contained within the module's compiled source code. This comprises the following primary components:

- SBC SWe application software
- Ribbon's proprietary ConnexIP operating system (with crypto libraries)
- SonusDB (with configuration files and PostGRES)
- HMAC digest files (for integrity checking)

The cryptographic boundary is the contiguous perimeter that surrounds all memory-mapped functionality provided by the module when loaded and stored in the host device's memory. The module is entirely contained within the physical perimeter.

Figure 2 shows the logical block diagram of the module executing in memory and its interactions with surrounding software components, as well as the module's physical perimeter and cryptographic boundary.

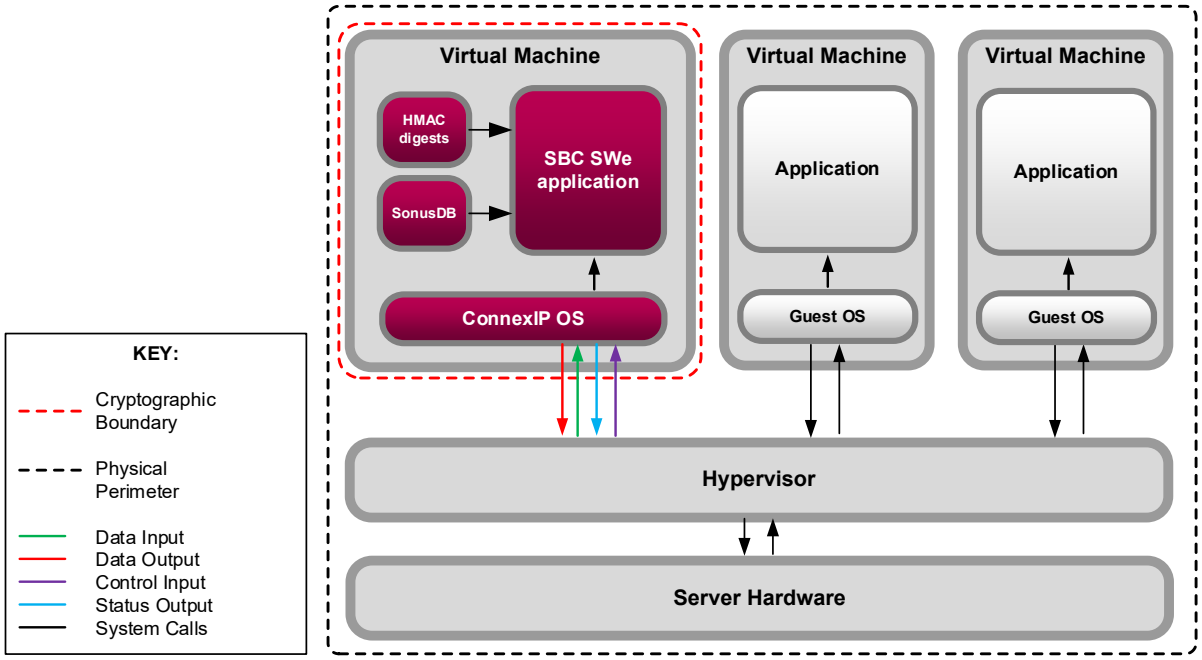

Figure 2. Module Block Diagram (with Cryptographic Boundary)

2.1.6 Tested Operational Environment's Physical Perimeter (TOEPP)

The physical perimeter of the cryptographic module is defined by the hard enclosure around the GPC on which it runs. Figure 3 is a diagram illustrating the hardware components of a GPC (the dashed line surrounding the hardware components represents the module's physical perimeter and identifies the hardware with which the processors interface.

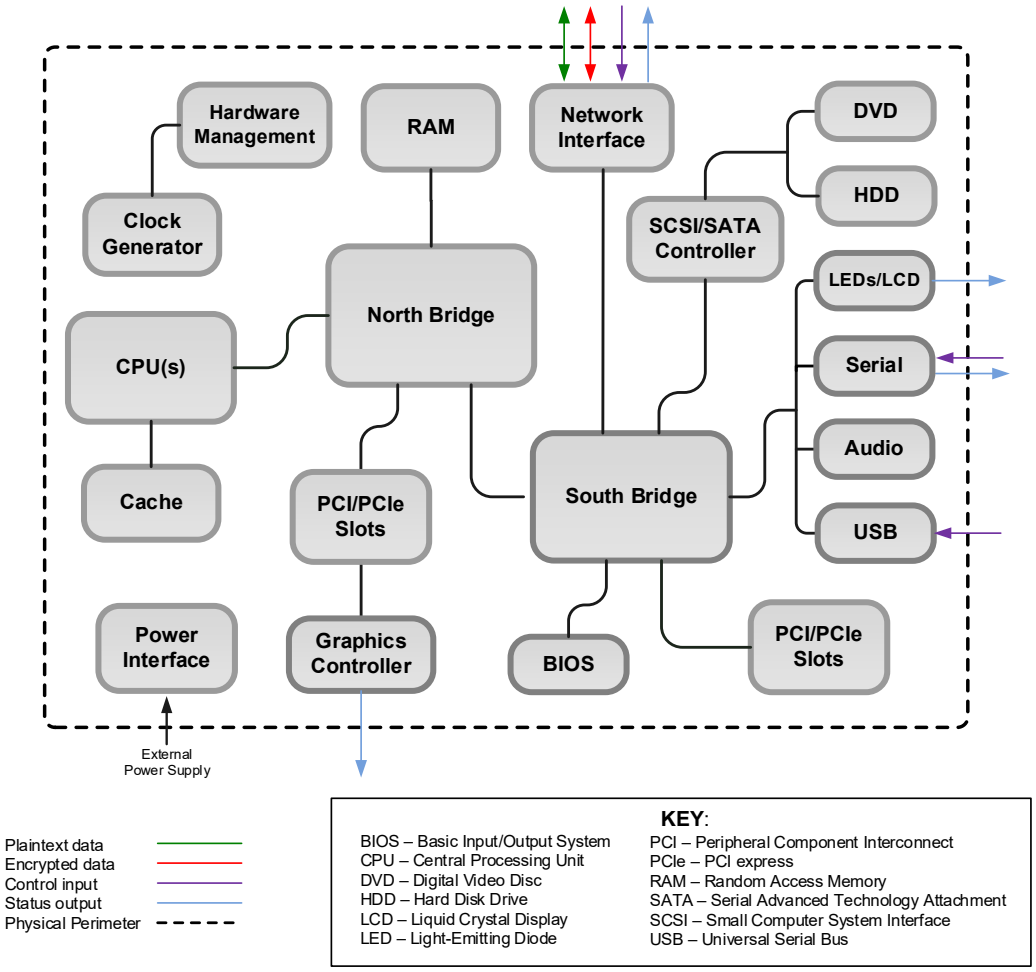

Figure 3. Block Diagram of a GPC

2.2 Tested and Vendor Affirmed Module Version and Identification

2.2.1 Tested Module Identification – Hardware

This section is only applicable for hardware modules.

N/A for this module.

2.2.2 Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

The module has the following executable code sets shown in the table below.

Package or File Name	Software/ Firmware Version	Features	Integrity Test
SBC SWe Session Border Controller	10.01.06		Yes

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

2.2.3 Tested Module Identification – Hybrid Disjoint Hardware

The module does not have hybrid disjoint hardware.

N/A for this module.

2.2.4 Tested Operational Environments – Software, Firmware, Hybrid

The module was tested and found to be compliant with FIPS 140-3 requirements on the environments listed in the table below.

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Ribbon ConnexIP OS 10	HPE ProLiant DL380 Gen9	Intel Xeon CPU E5 v3 (Haswell)	Yes	VMWare ESXi 6.5	10.01.06
Ribbon ConnexIP OS 10	HPE ProLiant DL380 Gen9	Intel Xeon CPU E5 v3 (Haswell)	No	VMWare ESXi 6.5	10.01.06

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

The module is designed to utilize AES-NI[16] extended instruction set when available by the host platform's CPU for processor algorithm acceleration (PAA) of its AES implementation.

2.2.5 Vendor-Affirmed Operational Environments – Software, Firmware, Hybrid

The cryptographic module maintains validation compliance when operating in a VM with ConnexIP as the guest OS on any compatible GPC using a KVM hypervisor to provide the virtualization layer. Note that the host GPC may be deployed on-prem or in any of the following supported cloud environments:

- OpenStack
- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure

Operating System	Hardware Platform
OpenStack	
Amazon Web Services	
Google Cloud Platform	
Microsoft Azure	

[16] AES-NI – Advanced Encryption Algorithm New Instructions

Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

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 not listed on the validation certificate.

2.3 Excluded Components

The module does not exclude any components from the requirements.

2.4 Modes of Operation

2.4.1 Modes List and Description

When installed, configured, and operated according to this Security Policy, the module supports the Approved mode of operation only; non-Approved operations are not supported.

Mode Name	Description	Type	Status Indicator
Approved	When the module is installed, initialized, and operated as directed in the Security Policy section 11.1, the module supports an Approved mode of operation only.	Approved	Global FIPS status indicator

Table 5: Modes List and Description

2.5 Algorithms

2.5.1 Approved Algorithms

The module employs cryptographic algorithm implementations from the following sources:

- Ribbon Cryptographic Library version 10 (Cert. A5061)
- Ribbon Entropy Library version 10 (Cert. A4087)
- Ribbon IKE KDF[17] Library version 10 (Cert. A5062)
- Ribbon SRTP[18] KDF Library (IPP[19]) version 10 (Cert. A5063)
- Ribbon SSH KDF Library version 10 (Cert. A5064)
- Ribbon TLS KDF Library version 10 (Cert. A5065)

The module implements the Approved algorithms listed in the table below.

Ribbon Cryptographic Library

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5061	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB1	A5061	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

[17] KDF – Key Derivation Function
[18] SRTP – Secure Real-Time Transport Protocol
[19] IPP – Intel® integrated Performance Primitives

Algorithm	CAVP Cert	Properties	Reference
AES-CFB128	A5061	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A5061	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A5061	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A5061	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 256	SP 800-38D
Counter DRBG	A5061	Prediction Resistance - No, Yes Mode - AES-128 Derivation Function Enabled - No, Yes	SP 800-90A Rev. 1
DSA KeyGen (FIPS186-4)	A5061	L - 2048, 3072 N - 224, 256	FIPS 186-4
ECDSA KeyGen (FIPS186-5)	A5061	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A5061	Curve - P-224, P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A5061	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A5061	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC-SHA-1	A5061	Key Length - Key Length: 112-65528 Increment 8	FIPS 198-1
HMAC-SHA2-224	A5061	Key Length - Key Length: 112-65528 Increment 8	FIPS 198-1
HMAC-SHA2-256	A5061	Key Length - Key Length: 112-65528 Increment 8	FIPS 198-1
HMAC-SHA2-384	A5061	Key Length - Key Length: 112-65528 Increment 8	FIPS 198-1
HMAC-SHA2-512	A5061	Key Length - Key Length: 112-65528 Increment 8	FIPS 198-1
KAS-ECC CDH-Component SP800-56Ar3 (CVL)	A5061	Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A5061	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A5061	Domain Parameter Generation Methods - FB, FC Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KTS-IFC	A5061	Modulo - 2048, 3072, 4096 Key Generation Methods - rsakpg1-basic Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Key Length - 768	SP 800-56B Rev. 2
PBKDF	A5061	Iteration Count - Iteration Count: 10-1000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
RSA KeyGen (FIPS186-5)	A5061	Key Generation Mode - probable Modulo - 2048 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A5061	Modulo - 2048 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A5061	Modulo - 2048 Signature Type - pkcs1v1.5	FIPS 186-5
SHA-1	A5061	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-224	A5061	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A5061	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A5061	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A5061	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 6: Approved Algorithms - Ribbon Cryptographic Library

Ribbon SRTP KDF Library (IPP)

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5063	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A5063	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A5063	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
HMAC-SHA-1	A5063	Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
KDF SRTP (CVL)	A5063	AES Key Length - 128, 192	SP 800-135 Rev. 1
SHA-1	A5063	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4

Table 7: Approved Algorithms - Ribbon SRTP KDF Library (IPP)

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyGen (FIPS186-4)	A5061	Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A5061	Curve - B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A5061	Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A5061	Curve - B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
RSA KeyGen (FIPS186-4)	A5061	Key Generation Mode - B.3.3 Modulo - 2048 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A5061	Signature Type - PKCS 1.5 Modulo - 2048	FIPS 186-4
RSA SigVer (FIPS186-4)	A5061	Signature Type - PKCS 1.5 Modulo - 1024, 2048	FIPS 186-4
TLS v1.2 KDF RFC7627 (CVL)	A5065	Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1

Table 8: Approved Algorithms -

Ribbon IKE KDF Library

Algorithm	CAVP Cert	Properties	Reference
KDF IKEv1 (CVL)	A5062	Authentication Method - Digital Signature, Pre-shared Key, Public Key Encryption Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 2048 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512 Preshared Key Length - Preshared Key Length: 64-512 Increment 8	SP 800-135 Rev. 1
KDF IKEv2 (CVL)	A5062	Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 224-4096 Increment 8 Derived Keying Material Length - Derived Keying Material Length: 160-4096 Increment 8 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1

Table 9: Approved Algorithms - Ribbon IKE KDF Library

Ribbon IKE KDF Library

Algorithm	CAVP Cert	Properties	Reference
KDF IKEv1 (CVL)	A5062	Authentication Method - Digital Signature, Pre-shared Key, Public Key Encryption Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 2048 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512 Preshared Key Length - Preshared Key Length: 64-512 Increment 8	SP 800-135 Rev. 1
KDF IKEv2 (CVL)	A5062	Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 224-4096 Increment 8 Derived Keying Material Length - Derived Keying Material Length: 160-4096 Increment 8 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1

Table 10: Approved Algorithms - Ribbon IKE KDF Library

Ribbon SSH KDF Library

Algorithm	CAVP Cert	Properties	Reference
KDF SSH (CVL)	A5064	Cipher - AES-128 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1

Table 11: Approved Algorithms - Ribbon SSH KDF Library

Ribbon TLS KDF Library

Algorithm	CAVP Cert	Properties	Reference
KDF TLS (CVL)	A5065	TLS Version - v1.2 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1

Table 12: Approved Algorithms - Ribbon TLS KDF Library

Ribbon Entropy Library

Algorithm	CAVP Cert	Properties	Reference
SHA3-256	A4087	Message Length - Message Length: 0-65536 Increment 8	FIPS 202

Table 13: Approved Algorithms - Ribbon Entropy Library

2.5.2 Vendor Affirmed Algorithms

The vendor affirms the following cryptographic security methods in the table below:

Name	Properties	Implementation	Reference
CKG	CKG:Symmetric	Ribbon Cryptographic Library	Per SP 800-133 Rev. 2, section 4.

Table 14: Vendor-Affirmed Algorithms

2.5.3 Non-Approved, Allowed Algorithms

The table below lists the non-Approved algorithms implemented by the module that are allowed for use in the Approved mode of operation.

N/A for this module.

2.5.4 Non-Approved, Allowed Algorithms with No Security Claimed

The module does not offer any non-Approved algorithms allowed in the Approved mode of operation with no security claimed.

N/A for this module.

2.5.5 Non-Approved, Not Allowed Algorithms

The module does not offer non-Approved algorithms not allowed in the Approved mode of operation.

N/A for this module.

2.6 Security Function Implementations

The table below lists the security function implementations for this module.

Name	Type	Description	Properties	Algorithms
DRBG for Random Password Generation	DRBG	DRBG for generating random passwords.		Counter DRBG: (A5061)
SHA for Password Storage	SHA	SHA2-512 for hashing stored password values.		SHA2-512: (A5061)
RSA SigVer for Certificate Loading	DigSig-SigVer	RSA digital signature verification, used for certificate loading.		SHA2-224: (A5061) SHA2-256: (A5061) SHA2-384: (A5061) SHA2-512: (A5061) RSA SigVer (FIPS186-5): (A5061) RSA SigVer (FIPS186-4): (A5061)
CKG for CDB Key Generation	CKG	CKG for the Config Database (CDB) Key, which is used for the encryption/decryption of RSA or ECDSA private keys and pre-shared secrets for RADIUS.		Counter DRBG: (A5061)

Name	Type	Description	Properties	Algorithms
IPsec IKE	KAS-Full	Key Agreement for IKE. Key exchange is performed using ECDH or DH. Public/private keys generation is performed using RSA.		AES-CBC: (A5061) AES-CTR: (A5061) AES-GCM: (A5061) DSA KeyGen (FIPS186-4): (A5061) HMAC-SHA-1: (A5061) HMAC-SHA2-224: (A5061) HMAC-SHA2-384: (A5061) HMAC-SHA2-512: (A5061) KAS-ECC CDH-Component SP800-56Ar3: (A5061) KAS-ECC-SSC Sp800-56Ar3: (A5061) KAS-FFC-SSC Sp800-56Ar3: (A5061) KDF IKEv1: (A5062) KDF IKEv2: (A5062) SHA-1: (A5061) SHA2-224: (A5061) SHA2-256: (A5061) SHA2-384: (A5061) SHA2-512: (A5061) Counter DRBG: (A5061) HMAC-SHA2-256: (A5061) ECDSA KeyGen (FIPS186-5): (A5061) ECDSA KeyVer (FIPS186-5): (A5061) RSA KeyGen (FIPS186-5): (A5061) RSA SigGen (FIPS186-5): (A5061) RSA SigVer (FIPS186-5): (A5061) ECDSA SigGen (FIPS186-5): (A5061) ECDSA SigVer (FIPS186-5): (A5061) ECDSA KeyGen (FIPS186-4): (A5061) ECDSA KeyVer (FIPS186-4): (A5061) ECDSA SigGen (FIPS186-4): (A5061) ECDSA SigVer (FIPS186-4): (A5061) RSA KeyGen (FIPS186-4): (A5061) RSA SigGen (FIPS186-4): (A5061) RSA SigVer (FIPS186-4): (A5061)

Name	Type	Description	Properties	Algorithms
SRTP	KAS-Full	Key Agreement for SRTP. Key exchange is performed using ECDH or DH. Public/private keys generation is performed using ECDSA or RSA.		AES-CBC: (A5063) AES-CTR: (A5063) AES-ECB: (A5063) Counter DRBG: (A5061) DSA KeyGen (FIPS186-4): (A5061) HMAC-SHA-1: (A5063) KAS-ECC-SSC Sp800-56Ar3: (A5061) KAS-FFC-SSC Sp800-56Ar3: (A5061) KAS-ECC CDH-Component SP800-56Ar3: (A5061) RSA KeyGen (FIPS186-5): (A5061) RSA SigGen (FIPS186-5): (A5061) RSA SigVer (FIPS186-5): (A5061) ECDSA KeyGen (FIPS186-5): (A5061) ECDSA KeyVer (FIPS186-5): (A5061) ECDSA SigGen (FIPS186-5): (A5061) ECDSA SigVer (FIPS186-5): (A5061) KDF SRTP: (A5063) SHA-1: (A5063) ECDSA KeyGen (FIPS186-4): (A5061) ECDSA KeyVer (FIPS186-4): (A5061) ECDSA SigGen (FIPS186-4): (A5061) ECDSA SigVer (FIPS186-4): (A5061) RSA KeyGen (FIPS186-4): (A5061) RSA SigGen (FIPS186-4): (A5061) RSA SigVer (FIPS186-4): (A5061)
AES for SNMPv3	BC-UnAuth	AES-CFB for SNMPv3 packet encryption/decryption.		AES-CFB1: (A5061) AES-CFB8: (A5061) AES-CFB128: (A5061)
HMAC for SNMPv3	MAC	HMAC for SNMPv3 packet authentication.		HMAC-SHA-1: (A5061) SHA-1: (A5061)

Name	Type	Description	Properties	Algorithms
SSH	KAS-Full	Key Agreement for SSH. Key exchange is performed using ECDH or DH. Public/private keys generation is performed using ECDSA or RSA.		AES-CBC: (A5061) AES-CTR: (A5061) AES-GCM: (A5061) Counter DRBG: (A5061) DSA KeyGen (FIPS186-4): (A5061) HMAC-SHA-1: (A5061) KAS-ECC CDH-Component SP800-56Ar3: (A5061) KAS-ECC-SSC Sp800-56Ar3: (A5061) KAS-FFC-SSC Sp800-56Ar3: (A5061) KDF SSH: (A5064) SHA-1: (A5061) ECDSA KeyGen (FIPS186-5): (A5061) ECDSA KeyVer (FIPS186-5): (A5061) ECDSA SigGen (FIPS186-5): (A5061) ECDSA SigVer (FIPS186-5): (A5061) RSA KeyGen (FIPS186-5): (A5061) RSA SigGen (FIPS186-5): (A5061) RSA SigVer (FIPS186-5): (A5061) ECDSA KeyGen (FIPS186-4): (A5061) ECDSA KeyVer (FIPS186-4): (A5061) ECDSA SigGen (FIPS186-4): (A5061) ECDSA SigVer (FIPS186-4): (A5061) RSA KeyGen (FIPS186-4): (A5061) RSA SigGen (FIPS186-4): (A5061) RSA SigVer (FIPS186-4): (A5061)

Name	Type	Description	Properties	Algorithms
TLS v1.2	KAS-Full	Key Agreement for TLS v1.2. Key exchange is performed using ECDH or DH. Public/private keys generation is performed using ECDSA or RSA.		AES-CBC: (A5061) AES-GCM: (A5061) Counter DRBG: (A5061) DSA KeyGen (FIPS186-4): (A5061) HMAC-SHA2-256: (A5061) HMAC-SHA2-384: (A5061) KAS-ECC CDH-Component SP800-56Ar3: (A5061) KAS-ECC-SSC Sp800-56Ar3: (A5061) KAS-FFC-SSC Sp800-56Ar3: (A5061) KDF TLS: (A5065) KTS-IFC: (A5061) TLS v1.2 KDF RFC7627: (A5065) ECDSA KeyGen (FIPS186-5): (A5061) ECDSA KeyVer (FIPS186-5): (A5061) ECDSA SigGen (FIPS186-5): (A5061) ECDSA SigVer (FIPS186-5): (A5061) RSA KeyGen (FIPS186-5): (A5061) RSA SigGen (FIPS186-5): (A5061) RSA SigVer (FIPS186-5): (A5061) ECDSA KeyGen (FIPS186-4): (A5061) ECDSA KeyVer (FIPS186-4): (A5061) ECDSA SigGen (FIPS186-4): (A5061) ECDSA SigVer (FIPS186-4): (A5061) RSA KeyGen (FIPS186-4): (A5061) RSA SigGen (FIPS186-4): (A5061) RSA SigVer (FIPS186-4): (A5061)
DRBG (Random bits)	DRBG	Get random bits from the DBRG.		Counter DRBG: (A5061)
PBKDF	PBKDF	PBKDF for decrypting certificates		PBKDF: (A5061)

Name	Type	Description	Properties	Algorithms
ECDSA or RSA SigVer for Public Key Certificate Authentication	DigSig-SigVer	ECDSA or RSA signature verification for public key certificate authentication.		ECDSA SigVer (FIPS186-4): (A5061) ECDSA SigVer (FIPS186-5): (A5061) RSA SigVer (FIPS186-5): (A5061) SHA-1: (A5061) SHA2-224: (A5061) SHA2-256: (A5061) SHA2-384: (A5061) Counter DRBG: (A5061) RSA SigVer (FIPS186-4): (A5061)
Entropy Source Conditioner	ENT-Cond	SHA3-256 entropy source conditioner		SHA3-256: (A4087)

Table 15: Security Function Implementations

2.7 Algorithm Specific Information

The following is algorithm-specific information related to the module:

- AES-GCM: The AES-GCM IV[20] is used in the following protocols:

    - For TLS v1.2, the module supports acceptable AES-GCM cipher suites from section 3.3.1 of *NIST SP 800-52rev2*. Per scenario 1 in *FIPS 140-3 IG* C.H, the mechanism for IV generation is compliant with *RFC 5288*. The counter portion of the IV is strictly increasing. When the IV exhausts the maximum number of possible values for a given session key, a failure in encryption will occur and a handshake to establish a new encryption key will be required. The module will then trigger a handshake to establish a new encryption key.

    - For SRTP, the AES-GCM IV is constructed at its entirety internally deterministically per section 8.2.1 of *NIST SP 800-38D*. In compliance with *RFC 7714*, the 96-bit IV is formed by first concatenating 16 bits of zeroes, the 32-bit Synchronization Source identifier, the 32-bit rollover counter, and the 16-octet sequence number. As described in scenario 3 of *FIPS 140-3 IG* C.H and, the (key, IV) collision probability does not exceed 2^{-32} for a given key distributed to one or more cryptographic modules.

  In the event that the module's power is lost and then restored, the CO shall establish a new key for use with AES-GCM encryption/decryption.

- PBKDF2: The module uses PBKDF2 option 1a from section 5.4 of *NIST SP 800-132*. This function takes an input salt that is 128 bits in length with a passphrase containing at least eight characters (in accordance with the module's password complexity requirements) and produces a random value of 128 bits. The underlying pseudorandom function used in this derivation is SHA-1.

  The function has an iteration count of 2048. This iteration count allows for user-acceptable key generation times while still doubling the recommended minimum count (1000).

[20] IV – Initialization Vector

The length of the password/passphrase used in the PBKDF shall be at least 20 characters, and shall consist of lowercase, uppercase, and numeric characters. The upper bound for the probability of guessing the value is estimated to be $1/62^{20} = 10^{-36}$, which is less than 2^{-112}.

As specified in *NIST SP 800-132*, keys derived from passwords/passphrases may only be used in storage applications.

- DH and ECDH: The module implements the DH and ECDH key agreement schemes specified in *NIST SP 800-56Arev3*. This specification requires that certain checks are performed to provide assurance regarding the keys being used. The following assurance checks are performed by the cryptographic module:
    - Assurances of domain parameter validity (section 5.5.2 of *NIST SP 800-56Arev3*)
    - Assurances required by the key pair owner (section 5.6.2.1 of *NIST SP 800-56Arev3*)
    - Assurances required by the public key recipient (section 5.6.2.2 of *NIST SP 800-56Arev3*)

- ECDSA: Per *FIPS 140-3 Implementation Guidance* C.K, there is currently no scheduled transition away from elliptic curves over binary fields (i.e., K-233, B-233, K-283, B-283, K-409, B-409, K-571, B-571). However, these curves are now deprecated, and it is strongly recommended to use the SP-800-186-defined prime curves (i.e., P-224, P-256, P-384, P-521) for the generation of the ECDSA signatures. Despite their deprecation status, these curves are still considered Approved.

2.8 RNG and Entropy

The module uses its Approved DRBG to generate cryptographic keys and seeds used to generate cryptographic keys. The resulting symmetric key or generated seed is an unmodified output from the DRBG.

The table below specifies the module's entropy certificates.

Cert Number	Vendor Name
E101	Ribbon Communications

Table 16: Entropy Certificates

The table below specifies the module's entropy sources.

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Ribbon Entropy Library	Non-Physical	Ribbon ConnexIP OS 10 on VMware ESXi 6.5 running on Intel Xeon CPU E5 v3 (Haswell)	64	Full entropy	SHA3-256 (A4087)

Table 17: Entropy Sources

2.9 Key Generation

The following is a list of key generation methods that the module implements:

- CKG
- ECDSA KeyGen
- RSA KeyGen

2.10 Key Establishment

2.10.1 Key Agreement Information

The following is a list of key agreement methods that the module implements:

- IKEv1 with DH or ECDH for Key Agreement
- IKEv2 with DH or ECDH for Key Agreement
- SRTP with DH or ECDH for Key Agreement
- SSH with DH or ECDH for Key Agreement
- TLS v1.2 with DH or ECDH for Key Agreement

The module implements the DH and ECDH key agreement schemes specified in *NIST SP 800-56Arev3*. This specification requires that certain checks are performed to provide assurance regarding the keys being used. The following assurance checks are performed by the cryptographic module:

- Assurances of domain parameter validity (section 5.5.2 of *NIST SP 800-56Arev3*)
- Assurances required by the key pair owner (section 5.6.2.1 of *NIST SP 800-56Arev3*)
- Assurances required by the public key recipient (section 5.6.2.2 of *NIST SP 800-56Arev3*)

Key confirmation is not supported by the module.

2.10.2 Key Transport Information

Key transport information does not apply to the module.

2.11 Industry Protocols

The module employs the following industry protocols:

- IPsec (IKEv1 and IKEv2)
- SRTP
- SSH
- TLS v1.2

The KDFs associated with these protocols shall only be used within the context of their respective protocols. No parts of these protocols, other than the Approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP.

3. Cryptographic Module Interfaces

3.1 Ports and Interfaces

The module supports the following logical interfaces:

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

As a virtual appliance, the module has no physical characteristics. Its interfaces are logical; the hypervisor provides virtualized ports and interfaces for the module that map to the host server's physical ports and interfaces. The module relies on the physical and electrical characteristics, manual controls, and physical indicators of the host server.

The module consists of an application and OS running in a virtual machine. The OS (ConnexIP) communicates directly with the hypervisor, which virtualizes ports and interfaces for the module. A mapping of the virtual ports and interfaces to the defined logical interfaces is provided in the table below. Note that the module does not output control information, and thus has no specified control output interface.

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	Encrypted management traffic over TLS (GUI) and SSH (CLI); user data over SSH; encrypted media and signaling traffic over TLS (GUI); redundancy synchronization traffic; operational/system status information
N/A	Data Output	Encrypted management traffic over TLS (GUI) and SSH (CLI); encrypted media and signaling traffic over TLS (GUI); redundancy synchronization traffic; operational/system status information
N/A	Control Input	Encrypted management traffic over TLS (GUI) and SSH (CLI); user data over SSH; encrypted media and signaling traffic over TLS (GUI); redundancy synchronization traffic; operational/system status information
N/A	Status Output	Encrypted management traffic over TLS (GUI) and SSH (CLI); encrypted media and signaling traffic over TLS (GUI); redundancy synchronization traffic; operational/system status information
N/A	Power	Power

Table 18: Ports and Interfaces

4. Roles, Services, and Authentication

4.1 Authentication Methods

Module operators are required to authenticate to the module for assumption of an authorized role. The module supports identity-based authentication. Role assumption is implicit, as module operator roles are assigned to their user account.

The module can support multiple operator sessions concurrently from multiple client devices. The maximum number of simultaneous sessions allowed per operator can be configured for any number from 1 to 5.

Each session remains active (logged in) and secured until the operator logs out or is automatically logged out from inactivity. When the module is powered off, result of any previous authentication will be cleared, and module operators will need to re-authenticate in order to re-assume their respective roles.

The table below lists the authentication methods and the strength of the authentication mechanisms.

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Password	For the first-time module access by the CO, the module ships with a factory-set default username ("admin") and password ("admin") for the EMA and the CLI. User accounts are assigned randomly generated passwords upon account activation. Following first-time login using default/assigned credentials, module operators are prompted to set new passwords.	Username/Password	1:16,889,161,502,720	6:16,889,161,502,720
Public Key Certificate	The module supports RSA and ECDSA digital certificate authentication of users during Web GUI/HTTPS (TLS) access.	ECDSA and RSA SigVer	1:5.19 x 10^33	1:96.92 × 10^23

Table 19: Authentication Methods

The strength objectives of the authentication mechanisms are as follows:

- For each attempt to use an authentication mechanism, the probability shall be less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur.
- For multiple attempts to use an authentication mechanism during a one-minute period, the probability shall be less than one in 100,000 that a random attempt will succeed or a false acceptance will occur.

To meet the stated strength objectives for password-based authentication, password policies shall be configured such that all passwords shall require:

- Between 8 and 24 characters
- At least one lowercase letter
- At least one uppercase letter

- At least one digit
- At least one special character

The Crypto Officer shall set the module's password policies after the initial setup of the module. Password policies can be set via the EMA or CLI.

- Using the CLI, password policies are configured using the following command:

```
% set system admin <SYSTEM NAME> passwordRules
```

- Using the EMA, password policies are configured by navigating to **Administration -> Users and Application Management -> Application Management** from the SBC main screen. The CO shall ensure that the checkbox for "Use Separate Password Rules for Administrators" is unchecked.

Once set, the module enforces the password policies on all subsequent attempts to change a password.

The CO shall also ensure that the default brute force password attack mitigation remains enabled. Per the default settings, the module will lock an account for 30 seconds after three (3) consecutive failed login attempts. The CO can view and/or configure the brute force password attack settings by navigating to the **Administration -> Users and Application Management -> Application Management** window from the Using the EMA SBC main screen. Using the CLI, the CO can view and/or configure the brute force password attack settings using the following command:

```
% show system admin <systemName> accountManagement bruteForceAttack
```

4.2 Roles

The module supports the following vendor-defined groups and capabilities:

- Administrator – Read-Write access to all commands and data spaces.

- Calea – Read-write access to Lawful Intercept tables and Read access to other tables. Only an Admin user can add or remove user from this group. Only one user named "Calea" is allowed in this group, and the "Calea" user cannot be part of any other group.

- Field Service – Read-write access to all commands and data spaces excluding some administrative functions only available to admin users.

- Guest – Read-only access to all commands and data spaces except commands that deal with user accounts, logging and audit controls, the TOD clock and sensitive administrative items. They do not have access to the Security Event logs and management audit logs.

- Operator – Read-write access to all commands and data spaces except commands that deal with user accounts, logging and audit controls, the TOD clock, and sensitive administrative items. They do not have access to the Security Event logs and management audit logs and cannot execute any commands stopping or starting these audit log services.

- Security Auditor – Read-only access to view the security logs and management audit logs. The commands executed by the Security Auditor are logged in the Management Audit log.

The module supports two defined roles that operators may assume:

- Crypto Officer – The CO is responsible for initializing the module for first use, which includes the configuration of passwords, public and private keys, and other SSPs. The CO is also responsible for the management of all keys and SSPs, including their zeroization, and is the only operator that can install and configure the module for Approved mode of operation. The CO has access to all User services. The CO role consists of the vendor-defined roles “Administrator”, “Calea”, “Field Service”, and “Operator”.

- User – The User has read-only privileges and can show the status and statistics of the module, show the current status of the module, and connect to the module remotely using HTTPS and SSH. Users can also change their own passwords. The User role consists of the vendor-defined roles “Guest” and “Security Auditor”.

The table below lists the supported roles.

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Identity	CO	Password Public Key Certificate
User	Identity	User	Password Public Key Certificate

Table 20: Roles

4.3 Approved Services

Descriptions of the services available are provided in the table below.

When configured according to the guidance in section 11, the module only executes in an Approved mode of operation. Thus, as allowed per section C.H of *FIPS 140-3 Implementation Guidance*, the module provides indicators for the use of Approved services through a combination of an explicit indication (via a global FIPS mode indicator) and an implicit indication (via the successful completion of the service).

The keys and Sensitive Security Parameters (SSPs) listed in the table indicate the type of access required using the following notation:

- G = Generate: The module generates or derives the SSP.
- R = Read: The SSP is read from the module (e.g., the SSP is output).
- 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.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Commission the module	Commission the module by following the Security Policy guidelines	n/a	n/a	none	None	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Manage SBC license	Installs the license to enable SBC features; delete or update license; view current license status	n/a	Command	Status output	None	Crypto Officer
Configure the SBC system	Define network interfaces and settings; set protocols; configure authentication information; define policies and profiles	n/a	Command and parameter	Command response, status output	None	Crypto Officer
Configure routing policy and control services	Configure IP network parameters and profiles for signaling, media, call routing, call services, zone, IP ACL rules, NTP and DNS servers	n/a	Command and parameter	Command response, status output	None	Crypto Officer
Configure Crypto Suite Profile	Select crypto suites for SRTP, SRTCP, and SIP communication	n/a	Command and parameter	Command response, status output	None	Crypto Officer
Configure Call Data Record (CDR)	Configure log file behavior	n/a	Command and parameter	Command response, status output	None	Crypto Officer
Perform backup and restore services	Back up the current system configuration; restore the system configuration from a backup file; import and export backup files	n/a	Command and parameter	Command response, status output	TLS v1.2	Crypto Officer - SNMPv3 Privacy Key: R,W - SNMPv3 Authentication Key: R,W
Manage users	Create, edit, and delete users; define user accounts and assign permissions.	n/a	Command and parameter	Command response, status output	DRBG for Random Password Generation SHA for Password Storage DRBG (Random bits) Entropy Source Conditioner	Crypto Officer - User password: R,W - DRBG Entropy Input String: G,E - DRBG Seed: G,E - DRBG 'V' Value: G,E - DRBG 'Key' Value: G,E
Manage user sessions	Terminate User sessions	n/a	Command and parameter	Command response, status output	SSH TLS v1.2	Crypto Officer - SSH Session Key: Z - TLS Session Key: Z
Change password	Modify existing login passwords	n/a	Command and parameter	Command response, status output	SHA for Password Storage	Crypto Officer - Crypto Officer password: W User - User password: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Load certificate	Load new certificates	Global FIPS status indicator	Command	Command response, status output	RSA SigVer for Certificate Loading PBKDF	Crypto Officer - Certificate Load Key: W - TLS Private Key: W - TLS Public Key: W - CA Public Key: W
Run script	Run a script file (a text file containing a list of CLI commands to execute in sequence)	Global FIPS status indicator	Command	Command response, status output	None	Crypto Officer
Perform on-demand self-tests	Perform self-tests on-demand using CLI/EMA command	n/a	Command	Command response, status output	None	Crypto Officer
Perform network diagnostics	Monitor connections (e.g., ping)	n/a	Command	Command response, status output	None	Crypto Officer User
Show system status	Show the system status (including Approved mode status)	n/a	Command	Command response, status output	None	Crypto Officer User
Show system versioning	Show operational status information about module components (including hardware type and application version)	n/a	Command	Command response, status output	None	Crypto Officer User
View Event Log	View event status messages	n/a	Command	Command response, status output	None	Crypto Officer
Zeroize keys	Zeroize all SSPs (using CLI/EMA command to zeroize)	Completion indicator (message/log)	Command	Command response, status output	None	Crypto Officer - Config Database (CBD) Key: Z User - Config Database (CBD) Key: Z
Perform keying of CDB Key	Generate CDB key	Global FIPS status indicator	Command and parameters	Command response, status output	CKG for CDB Key Generation	Crypto Officer - Config Database (CBD) Key: G
Reboot/Reset	Reboot or reset the module	Global FIPS status indicator	Command	Command response, status output	None	Crypto Officer - Config Database (CBD) Key: Z

Establish IPsec connection	Establish an IPsec session using IKE	Global FIPS status indicator	Command and parameters	Command response, status output	IPsec IKE	Crypto Officer - ECDH Private Key: G,E - ECDH Public Key: G,E - ECDH Peer Public Key: W,E - DH Private Key: G,E - DH Public Key: G,E - DH Peer Public Key: W,E - IKE RSA Private Key: G,E - IKE RSA Public Key: G,R - IKE Shared Secret: W,E - IKE SKEYID: G,E - IKE Pre-shared Secret: G,E - IKE Encryption Key: G,E - IKE Authentication Key: G,E - IPsec Shared Secret: G,E - IPsec Encryption Key: G,E - IPsec Authentication Key: G,E - AES GCM IV: G,E - IKE Key Derivation Key: G,E User - ECDH Private Key: G,E - ECDH Public Key: G,E - ECDH Peer Public Key: W,E - DH Private Key: G,E - DH Public Key: G,E - DH Peer Public Key: W,E - IKE RSA Private Key: G,E - IKE RSA Public Key: G,R - IKE Shared Secret: W,E - IKE SKEYID: G,E - IKE Pre-shared

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Secret: G,E - IKE Encryption Key: G,E - IKE Authentication Key: G,E - IKE Key Derivation Key: G,E - IPsec Shared Secret: G,E - IPsec Encryption Key: G,E - IPsec Authentication Key: G,E - AES GCM IV: G,E
Establish SRTP connection	Establish a SRTP session using SIP/TLS protocols	Global FIPS status indicator	Command and parameters	Command response, status output	SRTP	Crypto Officer - SRTP Master Key: W,E - SRTP Authentication Key: G - SRTP Session Key: G - AES GCM IV: G,E User - SRTP Master Key: W,E - SRTP Authentication Key: G - SRTP Session Key: G - AES GCM IV: G,E
Manage SNMPv3 services	Manage keys, trap targets, users, and traps for SNMPv3	Global FIPS status indicator	Command and parameters	Status output	None	Crypto Officer - SNMPv3 Privacy Key: W - SNMPv3 Authentication Key: W User - SNMPv3 Privacy Key: W - SNMPv3 Authentication Key: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Perform SNMPv3 services	Perform SNMPv3 services	Global FIPS status indicator	Command and parameters	Status output	AES for SNMPv3 HMAC for SNMPv3	Crypto Officer - SNMPv3 Privacy Key: E - SNMPv3 Authentication Key: W User - SNMPv3 Privacy Key: E - SNMPv3 Authentication Key: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Establish SSH connection	Establish SSH connection	Global FIPS status indicator	Command and parameters	Command response, status output	SSH	Crypto Officer - ECDH Private Key: G,E - ECDH Public Key: G,R - ECDH Peer Public Key: W,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - SSH Private Key: G,W,E - SSH Public Key: G,R - SSH Peer Public Key: W,E - SSH Session Key: G - SSH Authentication Key: G - SSH Shared Secret: G,E - AES GCM IV: G,E User - ECDH Private Key: G,E - ECDH Public Key: G,R - ECDH Peer Public Key: W,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - SSH Private Key: G,W,E - SSH Public Key: G,R - SSH Peer Public Key: W,E - SSH Session Key: G - SSH Authentication Key: G - SSH Shared Secret: G,E - AES GCM IV: G,E

Perform SFTP functions	Perform FTP functions over SSH	Global FIPS status indicator	Command and parameters	Command response, status output	SSH	Crypto Officer - ECDH Private Key: G,E - ECDH Public Key: G,R - ECDH Peer Public Key: W,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - SSH Private Key: G,W,E - SSH Public Key: G,R - SSH Peer Public Key: W,E - SSH Session Key: G - SSH Authentication Key: G - SSH Shared Secret: G,E - SFTP Private Key: G,E - SFTP Public Key: G,E - SFTP Peer Public Key: G,E - AES GCM IV: G,E User - ECDH Private Key: G,E - ECDH Public Key: G,R - ECDH Peer Public Key: W,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - SSH Private Key: G,W,E - SSH Public Key: G,R - SSH Peer Public Key: W,E - SSH Session Key: G,E - SSH Authentication Key: G,E - SSH Shared Secret: G,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SFTP Private Key: G,E - SFTP Public Key: G,E - SFTP Peer Public Key: G,E - AES GCM IV: G,E

Establish TLS connection	Establish TLS connection	Global FIPS status indicator	Command and parameters	Command response, status output	TLS v1.2	Crypto Officer - ECDH Private Key: G,E - ECDH Public Key: G,R - ECDH Peer Public Key: W,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - TLS Private Key: G,W,E - TLS Public Key: G,R,W - TLS Peer Public Key: W,E - TLS Session Key: G - TLS Authentication Key: G - TLS Pre-Master Secret: G,E - TLS Master Secret: G,E - AES GCM IV: G,E User - ECDH Private Key: G,E - ECDH Public Key: G,R - ECDH Peer Public Key: W,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - TLS Private Key: G,W,E - TLS Public Key: G,R,W - TLS Peer Public Key: W,E - TLS Session Key: G - TLS Authentication Key: G - TLS Pre-Master Secret: G,E - TLS Master Secret: G,E - AES GCM IV: G,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Logout	Logout of active session	n/a	Command	Command response, status output	None	Crypto Officer User
Perform manual zeroization	Zeroize ephemeral SSPs via power-cycle or reboot of host device	n/a	Command	Command response, status output	None	Crypto Officer
Perform manual on-demand self-tests	Perform pre-operational self-tests on demand via power-cycle or reboot of host device	n/a	Command	Command response, status output	None	Crypto Officer
Authenticate via password	Authenticate to the module by password	n/a	Command and parameters	Command response, status output	SHA for Password Storage	Crypto Officer - Crypto Officer password: W - RADIUS Shared Secret: W,E User - User password: W - RADIUS Shared Secret: W,E
Upgrade firmware	Load new firmware and performs an integrity test using an RSA digital signature	Global FIPS status indicator	Command, firmware image	Command response, status output	RSA SigVer for Certificate Loading	Crypto Officer - Image Verify Key: E
Authenticate via public key certificate	Authenticate to the module by ECDSA or RSA signature verification	n/a	Command	Command response, service access.	ECDSA or RSA SigVer for Public Key Certificate Authentication	Unauthenticated

Table 21: Approved Services

4.4 Non-Approved Services

The module does not provide any non-Approved services in the non-Approved mode of operation.

N/A for this module.

4.5 External Software/Firmware Loaded

The cryptographic module has the capability of loading software from an external source. All loads comprise a complete image replacement that replaces the existing module image in its entirety, thus forming a completely new module.

Upon load, the module will verify the integrity of the load package with a 2048-bit RSA signature verification (the public key is loaded in write-protected flash at the factory). If the check fails, the new software is ignored and the current software remains loaded. If successful, the module will force an automatic reboot and will perform the required pre-operational integrity test using HMAC SHA digests. Per *FIPS 140-3 IG* 10.3.F, the integrity test inherently meets the firmware load test requirement when the load constitutes a full image replacement.

Only validated software may be loaded to maintain the module's validation. The new image shall include updated versioning information to represent the newly loaded image.

Prior to any execution of the new image, module operators shall zeroize all persistent keys using one of the methods described in section 9.3. All ephemeral keys are zeroized on module reboot.

5. Software/Firmware Security

5.1 Integrity Techniques

At module start-up, all software components of the cryptographic module are verified using an approved integrity technique implemented within the cryptographic module itself. The module employs a series of HMAC SHA2-256 digests; unsuccessful verification of any of the digest values will cause the module to enter a critical error state.

5.2 Initiate on Demand

The CO can initiate the pre-operational integrity tests on demand by issuing a reset/reboot command over the module's management interfaces. Also, the module can be made to perform pre-operational self-tests by rebooting or power-cycling the module's VM manually (when using this method, the operator is not required to assume an authorized role).

6. Operational Environment

6.1 Operational Environment Type and Requirements

The SBC SWe Session Border Controller comprises a software cryptographic library that executes in a Modifiable operational environment. The module was tested and found to be compliant with FIPS 140-3 requirements on a Dell EMC PowerEdge r740 server with an Intel Xeon processor running Ribbon's ConnexIP OS as the guest OS in a VM managed by a VMware ESXi 7.0 hypervisor.

The cryptographic module has control over its own SSPs. The process and memory management functionality of the guest OS and the hypervisor 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 interfaces. The operational environment provides 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 or operators.

7. Physical Security

The SBC SWe is a multi-chip standalone software cryptographic module and does not include physical security mechanisms. Therefore, per *ISO/IEC 19790:2012* section 7.7.1, the requirements for physical security are not applicable.

8. Non-Invasive Security

This section is not applicable. There are currently no approved non-invasive mitigation techniques references in Annex F of *ISO/IEC 19790:2012*.

9. Sensitive Security Parameters Management

9.1 Storage Areas

The table below lists sensitive security parameters (SSPs) storage areas for this module. Section 9.4 selects from the storage areas listed and specifies the appropriate parameter in the “Storage” column if applicable to a specific SSP.

Storage Area Name	Description	Persistence Type
Flash Memory	Persistent storage of SSPs on flash memory	Static
RAM	Dynamic storage of SSPs on RAM	Dynamic
Spinning Media	Persistent storage of SSPs on spinning media	Static
SSD	Persistent storage of SSPs on SSD	Static
CDB on SSD	Persistent storage of SSPs in CDB on SSD	Static

Table 22: Storage Areas

9.2 SSP Input-Output Methods

The table below lists SSP input and output methods for this module. Section 9.4 selects from the input and output methods listed and specifies the appropriate parameter in the “Inputs/Outputs” column if applicable to a specific SSP.

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
External to RAM via DER file format	External	RAM	Plaintext	Manual	Electronic	
RAM to External via Encrypted Form	RAM	External	Encrypted	Automated	Electronic	TLS v1.2
External to RAM via Plaintext Certificate	External	RAM	Plaintext	Automated	Electronic	RSA SigVer for Certificate Loading
RAM to External via Plaintext	RAM	External	Plaintext	Automated	Electronic	TLS v1.2
External to CDB on SSD via CLI (SSH)	External	CDB on SSD	Encrypted	Manual	Electronic	SSH
External to CDB on SSD via EMA (TLS)	External	CDB on SSD	Encrypted	Manual	Electronic	TLS v1.2
RAM to External via Plaintext Certificate	RAM	External	Plaintext	Automated	Electronic	TLS v1.2
Backup to External via TLS	CDB on SSD	External	Encrypted	Automated	Electronic	TLS v1.2

Table 23: SSP Input-Output Methods

9.3 SSP Zeroization Methods

The table below lists SSP zeroization methods for this module. Section 9.4 selects from the zeroization methods listed and specifies the appropriate parameter in the “Zeroization” column if applicable to a specific SSP.

Zeroization Method	Description	Rationale	Operator Initiation
Command via CLI or EMA	Zeroization upon command via CLI or EMA	Zeroize all SSPs (using CLI/EMA command to zeroize)	CLI or EMA
Reboot	Zeroization upon rebooting the module	Reboot to zeroize RAM	Reboot
Session Termination	Upon Session Termination	Upon Session Termination to zeroize session based SSPs	Terminate session

Table 24: SSP Zeroization Methods

9.4 SSPs

The module supports the keys and other SSPs listed in the tables below. Note that all SSP imports and exports are electronic and performed within the TOEPP.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Config Database (CBD) Key	Encryption of RSA and ECDSA private keys and pre-shared secrets for RADIUS in CDB	192 - 192	Symmetric Key - CSP	DRBG (Random bits)		CKG for CDB Key Generation
CA Public Key	Verification of Certificate Authority signatures	2048 - 112 bits	Public Key - PSP			RSA SigVer for Certificate Loading
ECDH Private Key	Input to ECDH shared secret computation	Between 224 and 512 bits - Between 112 and 256 bits	Private Key - CSP	IPsec IKE SRTP SSH TLS v1.2		IPsec IKE SRTP SSH TLS v1.2
ECDH Public Key	Used by peer for ECDH shared secret computation	Between 224 and 512 bits - Between 112 and 256 bits	Public Key - PSP	IPsec IKE SRTP SSH TLS v1.2		IPsec IKE SRTP SSH TLS v1.2
ECDH Peer Public Key	Input to ECDH shared secret computation	Between 224 and 512 bits - Between 112 and 256 bits	Public Key - PSP		IPsec IKE SRTP SSH TLS v1.2	IPsec IKE SRTP SSH TLS v1.2
DH Private Key	Input to DH shared secret computation	2048 or 3072 bits - 112 or 128 bits	Private Key - CSP	IPsec IKE SRTP SSH TLS v1.2		IPsec IKE SRTP SSH TLS v1.2
DH Public Key	Used by peer for DH shared secret computation	2048 or 3072 bits - 112 or 128 bits	Public Key - PSP	IPsec IKE SRTP SSH TLS v1.2		IPsec IKE SRTP SSH TLS v1.2
DH Peer Public Key	Input to DH shared secret computation	2048 or 3072 bits - 112 or 128 bits	Public Key - PSP		IPsec IKE SRTP SSH TLS v1.2	IPsec IKE SRTP SSH TLS v1.2

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
IKE RSA Private Key	The key used in IKE authentication	2048 - 112 bits	Private Key - CSP	IPsec IKE		IPsec IKE
IKE RSA Public Key	The key used in IKE authentication	2048 - 112 bits	Public Key - PSP	IPsec IKE		IPsec IKE
IKE Pre-shared Secret	Used in the computation of the SKEYID during IKEv1 Phase 1	n/a - n/a	Shared Secret - CSP			IPsec IKE
IKE SKEYID	Used as input to IKEv1 KDF to derive IPsec Encryption Key and IPsec Authentication Key	112 minimum - 112 minimum	Keying material - CSP		IPsec IKE	IPsec IKE
IKE Shared Secret	Used as input to IKEv1 KDF to derive IPsec Encryption Key and IPsec Authentication Key	n/a - n/a	Shared Secret - CSP		IPsec IKE	IPsec IKE
IKE Encryption Key	Used during IKE Phase 2 to encrypt and decrypt its messages	128, 192, 256 - Between 128 and 256 bits	Secret Key - CSP	KDF IKEv1 (A5062)		IPsec IKE
IKE Authentication Key	Used during IKE Phase 2 to authenticate its messages	160 bits - 112 minimum	Symmetric Key - CSP	KDF IKEv1 (A5062)		IPsec IKE
IKE Key Derivation Key	Used by IKE KDFs for derivation of IPsec Encryption Key and IPsec Authentication Key	128, 192, 256 - Between 128 and 256 bits	Derivation Key - CSP	KDF IKEv1 (A5062)		IPsec IKE
IPsec Shared Secret	Used by IKE KDFs for derivation of IPsec Encryption Key and IPsec Authentication Key	n/a - n/a	Shared Secret - CSP		IPsec IKE	IPsec IKE
IPsec Encryption Key	Used during IPsec for traffic protection	128, 192, 256 - Between 128 and 256 bits	Secret Key - CSP	KDF IKEv1 (A5062) KDF IKEv2 (A5062)		IPsec IKE
IPsec Authentication Key	Used during IPsec for payload traffic verification	160 bits - 112 minimum	Secret Key - CSP	KDF IKEv1 (A5062) KDF IKEv2 (A5062)		IPsec IKE
SSH Private Key	Authentication during SSH session negotiation	Between 112 and 256 bits - Between 112 and 256 bits	Private Key - CSP	SSH		SSH
SSH Public Key	Authentication by peer during SSH session negotiation	Between 112 and 256 bits - Between 112 and 256 bits	Public Key - PSP	SSH		SSH
SSH Peer Public Key	Authentication during SSH session negotiation 1024-bit key is used for signature verification only	Between 112 and 256 bits - Between 112 and 256 bits	Public Key - PSP		SSH	
SSH Session Key	Encryption and decryption of SSH session packets	128 or 256 bits - 128 or 256 bits	Secret Key - CSP		SSH	SSH
SSH Authentication Key	Authentication of SSH session packets	160 bits - 112 minimum	Secret Key - CSP		SSH	SSH

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TLS Private Key	RSA/ECDSA certificate-based authentication during TLS key negotiation	Between 112 and 256 bits - Between 112 and 256 bits	Private Key - CSP	TLS v1.2		TLS v1.2
TLS Public Key	RSA/ECDSA certificate-based authentication during TLS key negotiation	Between 112 and 256 bits - Between 112 and 256 bits	Public Key - PSP	TLS v1.2		TLS v1.2
TLS Peer Public Key	RSA/ECDSA certificate-based authentication during TLS key negotiation	Between 112 and 256 bits - Between 112 and 256 bits	Public Key - PSP		TLS v1.2	TLS v1.2
TLS Session Key	Encryption and decryption of TLS session packets	128 or 256 bits - Between 128 or 256 bits	Secret Key - CSP		TLS v1.2	TLS v1.2
TLS Authentication Key	Authentication of TLS session packets	160 bits (minimum) - 112 minimum	Secret Key - CSP		TLS v1.2	TLS v1.2
SRTP Session Key	Encryption or decryption of SRTP session packets	128 or 256 bits - Between 128 or 256 bits	Secret Key - CSP		SRTP	SRTP
SRTP Authentication Key	Authentication of SRTP session packets	160 bits - 112 minimum	Secret Key - CSP		SRTP	SRTP
SFTP Private Key	Used for SFTP key negotiation	2048 - 112 bits	Private Key - CSP	SSH		SSH
SFTP Public Key	Used for SFTP key negotiation	1024 or 2048 bits - 112 bits	Public Key - PSP	SSH		SSH
SFTP Peer Public Key	Used for SFTP key negotiation	1024 or 2048 bits - 80 or 112 bits	Public Key - PSP		SSH	SSH
SNMPv3 Privacy Key	Encrypting SNMPv3 packets	128 bits - 128 bits	Secret Key - CSP			
SNMPv3 Authentication Key	SNMPv3 Authentication Key	160 bits (minimum) - 112 minimum	Secret Key - CSP			
Certificate Load Key	Decrypting PKCS #12 certificate files when imported from an external workstation	128 or 256 bits - Between 128 or 256 bits	Secret Key - CSP		PBKDF	RSA SigVer for Certificate Loading
SRTP Master Key	Peer Authentication; Input to SRTP KDF for derivation of the SRTP Session Key and SRTP Authentication Key	n/a - n/a	Secret Key - CSP			
SSH Shared Secret	Input to SSH KDF for derivation of the SSH Session Key and SSH Authentication Key	n/a - n/a	Shared Secret - CSP		SSH	SSH
TLS Pre-Master Secret	Input to TLS KDF for derivation of the TLS Master Secret	n/a - n/a	Pre-Master Secret - CSP	DRBG (Random bits)	TLS v1.2	TLS v1.2
TLS Master Secret	Used by TLS KDF for derivation of the TLS Session Key and TLS Authentication Key	n/a - n/a	Master Secret - CSP		TLS v1.2	TLS v1.2

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
RADIUS Shared Secret	Peer authentication of RADIUS messages	n/a - n/a	Shared Secret - CSP			
DRBG Entropy Input String	Establishment of seed for CTR_DRBG	128 - 128	Entropy Input - CSP			DRBG (Random bits)
DRBG Seed	Generation of random number	256 - 256	Seed - CSP			DRBG (Random bits)
DRBG 'V' Value	State value for CTR_DRBG	128 - 128	Working state value - CSP		DRBG (Random bits)	
DRBG 'Key' Value	State value for CTR_DRBG	128 - 128	Working state value - CSP			
Crypto Officer password	Authenticating the Crypto Officer to the module	n/a - n/a	Password - CSP			
User password	Authenticating the User to the module	n/a - n/a	Password - CSP			
AES GCM IV	Initialization vector for the AES GCM key	96 bits - 112 bits	Initialization Vector - CSP	IPsec IKE SRTP SSH TLS v1.2		IPsec IKE SRTP SSH TLS v1.2
Image Verify Key	Verifying the RSA signature of the digest of a new image load package	2048 bits - 112 bits	Public Key - Neither			RSA SigVer for Certificate Loading

Table 25: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Config Database (CBD) Key		SSD:Plaintext		Command via CLI or EMA	
CA Public Key	External to RAM via DER file format	RAM:Plaintext	Until reboot	Reboot	
ECDH Private Key		RAM:Plaintext	Session	Reboot Session Termination	
ECDH Public Key	RAM to External via Plaintext Certificate	RAM:Plaintext	Session	Reboot Session Termination	
ECDH Peer Public Key	External to RAM via Plaintext Certificate	RAM:Plaintext	Session	Reboot Session Termination	
DH Private Key		RAM:Plaintext	Session	Reboot Session Termination	
DH Public Key	RAM to External via Plaintext	RAM:Plaintext	Session	Reboot Session Termination	
DH Peer Public Key	External to RAM via Plaintext Certificate	RAM:Plaintext	Session	Reboot Session Termination	
IKE RSA Private Key		SSD:Encrypted	Until zeroized	Command via CLI or EMA	
IKE RSA Public Key		SSD:Encrypted	Until zeroized	Command via CLI or EMA	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
IKE Pre-shared Secret	External to CDB on SSD via CLI (SSH) External to CDB on SSD via EMA (TLS)	RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
IKE SKEYID		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
IKE Shared Secret		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
IKE Encryption Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
IKE Authentication Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
IKE Key Derivation Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
IPsec Shared Secret		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
IPsec Encryption Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
IPsec Authentication Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
SSH Private Key		SSD:Plaintext		Command via CLI or EMA	
SSH Public Key		SSD:Plaintext		Command via CLI or EMA	
SSH Peer Public Key	External to RAM via Plaintext Certificate	RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
SSH Session Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
SSH Authentication Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
TLS Private Key	External to RAM via DER file format	SSD:Encrypted		Command via CLI or EMA	
TLS Public Key	External to RAM via Plaintext Certificate	SSD:Encrypted		Command via CLI or EMA	
TLS Peer Public Key	External to RAM via Plaintext Certificate	RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
TLS Session Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
TLS Authentication Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SRTP Session Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
SRTP Authentication Key		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
SFTP Private Key		SSD:Encrypted		Command via CLI or EMA	
SFTP Public Key	External to RAM via Plaintext Certificate	SSD:Plaintext		Command via CLI or EMA	
SFTP Peer Public Key	External to RAM via Plaintext Certificate	RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
SNMPv3 Privacy Key	External to CDB on SSD via CLI (SSH) External to CDB on SSD via EMA (TLS) Backup to External via TLS	SSD:Encrypted		Command via CLI or EMA	
SNMPv3 Authentication Key	External to CDB on SSD via CLI (SSH) External to CDB on SSD via EMA (TLS) Backup to External via TLS	SSD:Encrypted		Command via CLI or EMA	
Certificate Load Key		RAM:Plaintext	Until reboot	Reboot	
SRTP Master Key	RAM to External via Encrypted Form	RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
SSH Shared Secret		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
TLS Pre-Master Secret		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
TLS Master Secret		RAM:Plaintext	Until reboot or session termination	Reboot Session Termination	
RADIUS Shared Secret	External to CDB on SSD via CLI (SSH) External to CDB on SSD via EMA (TLS)	SSD:Encrypted		Command via CLI or EMA	
DRBG Entropy Input String		RAM:Plaintext	Until reboot	Reboot	
DRBG Seed		RAM:Plaintext	Until reboot	Reboot	
DRBG 'V' Value		RAM:Plaintext	Until reboot	Reboot	
DRBG 'Key' Value		RAM:Plaintext	Until reboot	Reboot	
Crypto Officer password	External to CDB on SSD via CLI (SSH) External to CDB on SSD via EMA (TLS)	SSD:Encrypted		Command via CLI or EMA	
User password	External to CDB on SSD via CLI (SSH) External to CDB on SSD via EMA (TLS)	SSD:Encrypted		Command via CLI or EMA	
AES GCM IV		RAM:Plaintext	Until reboot	Reboot	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Image Verify Key	RAM to External via Encrypted Form External to RAM via Plaintext Certificate	Flash Memory:Plaintext Spinning Media:Plaintext		N/A	

Table 26: SSP Table 2

9.5 Transitions

The following list specifies applicable transition periods or timeframes where an algorithm or key length transitions from Approved to non-Approved:

- SHA-1: The module includes an implementation of SHA-1 for hashing and digital signature verification. This implementation will be non-Approved for all uses starting January 1, 2031.

10. Self-Tests

The module performs pre-operational self-tests and conditional self-tests. Pre-operational tests are performed between the time the cryptographic module is instantiated and before the module transitions to the operational state. Conditional self-tests are performed by the module during module operation when certain conditions exist. The following sections list the self-tests performed by the module, their expected error status, and the error resolutions.

10.1 Pre-Operational Self-Tests

The module performs the following pre-operational self-test(s):

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A5061)		Integrity test	SW/FW Integrity	log messages	software integrity test

Table 27: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

The module performs the following conditional self-tests:

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A5061)	256 bits	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Encrypt/Decrypt	Performed during module's initial power-up sequence.
Counter DRBG (A5061)	-	instantiate/generate/reseed KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	DRBG	Performed during module's initial power-up sequence.
DSA KeyGen (FIPS186-4) (A5061)	2048-bit	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Sign/Verify	Performed during module's initial power-up sequence.
ECDSA SigGen (FIPS186-5) (A5061)	curve P-224/K-233, SHA2-512	digital signature generation KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Signature Generation	Performed during module's initial power-up sequence.
ECDSA SigVer (FIPS186-4) (A5061)	curve P-224/K-233, SHA2-512	digital signature generation KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Signature Verification	Performed during module's initial power-up sequence.
Entropy Stuck Test	-	Entropy Stuck Test	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	SP800-90B tests	Performed during module's initial power-up sequence.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Entropy Repetition Stuck Test	-	Entropy Repetition Stuck Test	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	SP800-90B tests	Performed during module's initial power-up sequence.
Entropy Adaptive Proportion Test	-	Entropy Adaptive Proportion Test	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	SP800-90B tests	Performed during module's initial power-up sequence.
Entropy Lag Test	-	Entropy Lag Test	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	SP800-90B tests	Performed during module's initial power-up sequence.
HMAC-SHA-1 (A5061)	128 bits	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Hashed Message Authentication	Performed during module's initial power-up sequence.
HMAC-SHA2-224 (A5061)	128 bits	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Hashed Message Authentication	Performed during module's initial power-up sequence.
HMAC-SHA2-256 (A5061)	256 bits	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Hashed Message Authentication	Performed during module's initial power-up sequence.
HMAC-SHA2-384 (A5061)	192 bits	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Hashed Message Authentication	Performed during module's initial power-up sequence.
HMAC-SHA2-512 (A5061)	256 bits	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Hashed Message Authentication	Performed during module's initial power-up sequence.
KAS-FFC-SSC Sp800-56Ar3 (A5061)	"Z" computation	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Signature Verification	Performed during module's initial power-up sequence.
KAS-ECC-SSC Sp800-56Ar3 (A5061)	"Z" computation	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Primitive 'Z' Computation	Performed during module's initial power-up sequence.
RSA SigGen (FIPS186-5) (A5061)	2048-bit, SHA2-256, PKCS #1	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Signature Generation	Performed during module's initial power-up sequence.
RSA SigVer (FIPS186-5) (A5061)	2048-bit, SHA2-256, PKCS #1	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Signature Verification	Performed during module's initial power-up sequence.
RSA KeyGen (FIPS186-4) (A5061)	-	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface	Encrypt/Decrypt	Performed during module's initial power-up sequence.
SHA3-256 (A4087)	256 bits	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Hashed Message Authentication	Performed during module's initial power-up sequence.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF SRTP (A5063)	-	STRP KDF KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	STRP KDF Test	Performed during module's initial power-up sequence.
HMAC-SHA-1 (A5063)	128 bits	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Hashed Message Authentication	Performed during module's initial power-up sequence.
AES-CBC (A5063)	128 bits	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Encrypt/Decrypt	Performed during module's initial power-up sequence.
KDF IKEv1 (A5062)	-	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Key Derivation	Performed during module's initial power-up sequence.
KDF SSH (A5064)	-	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Key Derivation	Performed during module's initial power-up sequence.
KDF TLS (A5065)	-	KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Key Derivation	Performed during module's initial power-up sequence.
ECDSA KeyGen (FIPS186-5) (A5061)	P-256	PCT	PCT	Status can be displayed on CLI terminal of Web GUI Interface.	Sign/Verify	Performed during module's initial power-up sequence.
RSA KeyGen (FIPS186-5) (A5061)	2048-bit, SHA2-256	PCT	PCT	On Demand	Sign/Verify and Encrypt/Decrypt	Performed during module's initial power-up sequence.
DH Key Generation	-	PCT	PCT	Status can be displayed on CLI terminal of Web GUI Interface.	Key Generation	Conditional before first use.
ECDH Key Generation	-	PCT	PCT	Status can be displayed on CLI terminal of Web GUI Interface.	Key Generation	Conditional before first use.
PBKDF (A5061)		KAT	CAST	Status can be displayed on CLI terminal of Web GUI Interface.	Key Derivation	Performed during module's initial power-up sequence.

Table 28: Conditional Self-Tests

10.3 Periodic Self-Test Information

The operator may conduct on demand by rebooting or power-cycling the module's VM or host server. On-demand self-test information for pre-operational and conditional self-tests are shown in the tables below.

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A5061)	Integrity test	SW/FW Integrity	on module start up	on module reboot

Table 29: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (A5061)	KAT	CAST	On Demand	Manual
Counter DRBG (A5061)	instantiate/generate/reseed KAT	CAST	On Demand	Manual
DSA KeyGen (FIPS186-4) (A5061)	KAT	CAST	On Demand	Manual
ECDSA SigGen (FIPS186-5) (A5061)	digital signature generation KAT	CAST	On Demand	Manual
ECDSA SigVer (FIPS186-4) (A5061)	digital signature generation KAT	CAST	On Demand	Manual
Entropy Stuck Test	Entropy Stuck Test	CAST	On Demand	Manual
Entropy Repetition Stuck Test	Entropy Repetition Stuck Test	CAST	On Demand	Manual
Entropy Adaptive Proportion Test	Entropy Adaptive Proportion Test	CAST	On Demand	Manual
Entropy Lag Test	Entropy Lag Test	CAST	On Demand	Manual
HMAC-SHA-1 (A5061)	KAT	CAST	On Demand	Manual
HMAC-SHA2-224 (A5061)	KAT	CAST	On Demand	Manual
HMAC-SHA2-256 (A5061)	KAT	CAST	On Demand	Manual
HMAC-SHA2-384 (A5061)	KAT	CAST	On Demand	Manual
HMAC-SHA2-512 (A5061)	KAT	CAST	On Demand	Manual
KAS-FFC-SSC Sp800-56Ar3 (A5061)	KAT	CAST	On Demand	Manual
KAS-ECC-SSC Sp800-56Ar3 (A5061)	KAT	CAST	On Demand	Manual
RSA SigGen (FIPS186-5) (A5061)	KAT	CAST	On Demand	Manual
RSA SigVer (FIPS186-5) (A5061)	KAT	CAST	On Demand	Manual
RSA KeyGen (FIPS186-4) (A5061)	KAT	CAST	On Demand	Manual
SHA3-256 (A4087)	KAT	CAST	On Demand	Manual
KDF SRTP (A5063)	STRP KDF KAT	CAST	On Demand	Manual
HMAC-SHA-1 (A5063)	KAT	CAST	On Demand	Manual
AES-CBC (A5063)	KAT	CAST	On Demand	Manual
KDF IKEv1 (A5062)	KAT	CAST	On Demand	Manual
KDF SSH (A5064)	KAT	CAST	On Demand	Manual
KDF TLS (A5065)	KAT	CAST	On Demand	Manual
ECDSA KeyGen (FIPS186-5) (A5061)	PCT	PCT	On Demand	Manual
RSA KeyGen (FIPS186-5) (A5061)	PCT	PCT	On Demand	Manual
DH Key Generation	PCT	PCT	On Demand	Manual
ECDH Key Generation	PCT	PCT	On Demand	Manual
PBKDF (A5061)	KAT	CAST	On Demand	Manual

Table 30: Conditional Periodic Information

10.4 Error States

The table below describes the error states and status indicators of the module.

Name	Description	Conditions	Recovery Method	Indicator
Soft Error	Upon failure of the conditional firmware load test	Failure of the conditional firmware load test	Reject firmware load operation.	log file entry / console message
Critical Error	Upon failure of any other pre-operational self-test or conditional self-test	Failure of any other pre-operational self-test or conditional self-test	Reboot or reload module	log file entry / console message

Table 31: Error States

10.5 Operator Initiation of Self-Tests

The operator may initiate self-tests on demand by issuing the reboot command or power cycling the module's VM or host server.

11. Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

11.1.1 Secure Installation

The module is in an unconfigured operational state upon the initial power-up. The Crypto Officer shall be responsible for all initial setup activities, including configuring the virtual machine, installing the guest operating system, and installing the SBC SWe application software. For detailed guidance regarding these activities, please see the SBC Core 10.1.x Documentation webpage on Ribbon's online Documentation and Support Portal and refer to the following document entries:

- EMA User Guide
- CLI Reference Guide
- Installing SBC SWe on Virtual and Private Cloud Environments

The SBC SWe Session Border Controller is available as a software package that includes both the application software and the operating system. It can be downloaded from the Ribbon Support Portal (login credentials are required for portal access).

The SBC SWe must be installed on a VMware-based virtual environment. The CO must follow the applicable guidance found under the document entry "Installing SBC SWe on VMware", which includes:

- creating the virtual machine
- uploading the SBC software
- installing the SBC OS
- installing the SBC application in a standalone or high availability configuration

The next step is to configure the module's application software. The CO must follow the instructions under the document entry "SBC SWe Configuration Procedures", which provides detailed guidance for setting deployment-specific system settings and configuring various module features.

Once the module is installed with network settings properly configured, the Crypto Officer must then initialize the module into Approved mode.

11.1.2 Initialization

Prior to initializing the module for operation in the Approved mode, SNMPv3 must be reconfigured. All trap targets with securityLevel set for authPriv and authNoPriv must be disabled. The following steps using the CLI provide an example:

```
% show oam snmp trapTarget EMS_-10.54.71.176
ipAddress 10.54.71.176;
port 162;
trapType v3;
targetUsername emstrapuser;
```

```
targetSecurityLevel authPriv;
state enabled;

% set oam snmp trapTarget EMS_-10.54.71.176 state disabled
% commit
```

To initialize the module for operation in the Approved mode, the CO may use the CLI or the EMA. The sections below describe these initialization methods.

11.1.2.1 Initialization using CLI

When using the CLI, the CO shall complete the following procedure:

1. Log in to the CLI using the default username "admin" and password "admin".
2. Switch to "configure private" mode using the following command:

```
> configure private
```

3. Execute the following commands:

```
% set profiles security tlsProfile defaultTlsProfile v1_0 disabled v1_1 disabled v1_2 enabled
% set profiles security EmaTlsProfile defaultEmaTlsProfile v1_0 disabled v1_1 disabled v1_2 enabled
% set profiles security ikeProtectionProfile AesSha1IkeProfile algorithms dhGroup modp2048
% set oam snmp version v3only
% set system admin <system_name> fips-140-3 mode enabled
% commit
```

NOTE: Once the "fips-140-3 mode" is set to 'enabled', it cannot be disabled through the configuration. A fresh software installation is required to set the operational mode back to 'disabled'.

Setting "fips-140-3 mode" to 'enabled' accomplishes the following:

- regenerates all SSH keys.
- regenerates encryption keys used by the system configuration database.
- zeroizes all persistent CSPs from the system and causes the server to reboot after confirmation.

11.1.2.2 Initialization using EMA + CLI

The EMA does not include all of the commands necessary to enable/disable the Approved mode. The user must use the CLI to complete the procedure. When using the EMA and CLI, the CO shall complete the following procedure:

1. Log in to the EMA using the default username "admin" and password "admin".
2. Navigate to **All -> Profiles -> Security -> TLS Profile**. The **TLS Profile** window is displayed, with the **TLS Profile List** pane.
3. Select the radio button corresponding to the defaultTlsProfile. The **Edit Selected TLS Profile** pane is displayed.
4. Set the fields V1_0 and V1_1 to "Disabled". Set the field V1_2 to "Enabled". Click **Save** to save the changes.
5. Navigate to **All -> Profiles -> Security -> EMA TLS Profile**. The **EMA TLS Profile** window is displayed, with the **EMA TLS Profile List** pane.
6. Select the radio button corresponding to the defaultEmaTlsProfile. The **Edit Selected EMA TLS Profile** pane is displayed.
7. Set the fields V1_0 and V1_1 to "Disabled". Set the field V1_2 to "Enabled". Click **Save** to save the changes.
8. Navigate to **All -> OAM -> Snmp**. The **Snmp** window is displayed, with the **Edit Snmp** pane.

9. Set the Version field to “V3only”. Click **Save** to save the changes.
10. Log in to the CLI and execute the following commands:

```
% set system admin <system_name> fips-140-3 mode enabled
% commit
```

Setting “fips-140-3 mode” to ‘enabled’ accomplishes the following:

- regenerates all SSH keys.
- regenerates encryption keys used by the system configuration database.
- zeroizes all persistent CSPs from the system and causes the server to reboot after confirmation.

11.1.3 Startup

After completion and confirmation of the above steps, the module’s VM will reboot. After this reboot, and on all subsequent reboots, the module will be operating in its Approved mode.

The following steps are required for module startup.

11.1.3.1 Restoring EMA in Platform Mode

To restore service to the EMA in Platform Mode after “fips-140-3 mode” is set to ‘enabled’, CA certificates and newly-generated SBC certificates must be imported using the CLI.

- Import CA Certificates – Use the following procedure to import up to twenty CA certificates and associate them with the EmaTlsProfile named "defaultEmaTlsProfile."

```
> configure private
% set system security pki certificate intermediateCaCert fileName intCaCert.der state enabled type
remote
% set system security pki certificate rootCaCert fileName rootCaCert.der state enabled type remote
% commit
% set profiles security EmaTlsProfile defaultEmaTlsProfile ClientCaCert intermediateCaCert
% set profiles security EmaTlsProfile defaultEmaTlsProfile ClientCaCert rootCaCert
% commit
```

- Import SBC Certificates – The SBC enables importing SBC server certificates generated with either of two different methods: those generated externally and those generated locally in the SBC.

  Use the following procedure to import an externally-generated SBC key and certificate in PKCS#12 format:

  1. Transfer the PKCS#12 formatted key/certificate file to the SBC and save it as `opt/sonus/external/<filename>.p12`.
  2. Install the certificate using the following steps (the following example uses a certificate named "sbxCert.p12" with a passPhrase "sonus"):

```
> configure private
% set system security pki certificate sbxCert fileName sbxCert.p12 passPhrase sonus state
enabled type
local
% commit
% set profiles security EmaTlsProfile defaultEmaTlsProfile serverCertName sbxCert
% commit
```

Use the following procedure to generate an SBC key and CSR locally in the SBC, and then import as a PEM externally-signed certificate: The CO shall ensure that only keys providing at least 112 bits of encryption strength are used for signing certificate requests.

1. Generate a CSR:

```
> configure private
% set system security pki certificate sbxCert type local-internal
% commit
% exit

> request system security pki certificate sbxCert generateCSR keySize keySize2K csrSub
"/C=US/ST=MA/L=Westford/O=Sonus Networks Inc./CN=www.sonusnet.com"
```

2. Copy the CSR output from the request in step 1 and obtain a signed certificate in a PEM formatted file from the appropriate Certificate Authority.
3. Transfer the certificate to the SBC and save it as `/opt/sonus/external/<filename>.pem`.
4. Install the certificate using the following steps (the following example uses the certificate file name "sbxCert.pem"):

```
> configure private
% set system security pki certificate sbxCert fileName sbxCert.pem
% commit
% set profiles security EmaTlsProfile defaultEmaTlsProfile serverCertName sbxCert
% commit
```

11.1.3.2 Reconfiguring SNMP Keys

After “fips-140-3 mode” is set to ‘enabled’, the SNMP Authentication Keys and SNMPv3 Privacy Keys must be reconfigured for all SNMPv3 users (this applies to all SNMPv3 users for authPriv/authNoPriv security level trap targets).

1. Use the following CLI commands to reconfigure the keys:

```
% set oam snmp users <username> authKey <colon separated hex string>
% set oam snmp users <username> privKey <colon separated hex string>
% commit
```

2. Enable the authPriv/authNoPriv trap targets:

```
% set oam snmp trapTarget <trap_target_name> state enabled
% commit
```

11.2 Administrator Guidance

The Crypto Officer is responsible for initialization and security-relevant configuration and management of the module. Once the module is properly configured, the Crypto Officer is responsible for maintaining and monitoring the status of the module to ensure that it is running in its Approved mode. For details regarding the general management of the module, please refer to the appropriate entries on Ribbon’s SBC Core 10.1.x Documentation webpage.

11.2.1 Status Information

Operational mode status of the module can be viewed by navigating to **Administration** -> **Users and Application Management** -> **Fips-140-3** from the SBC main screen via the EMA and selecting the desired SBC system from the dropdown menu. When running in Approved mode, the radio button marked “Enabled” will be selected.

The Crypto Officer shall monitor the module’s status regularly. If any irregular activity is noticed, or the module is consistently reporting errors, customers should contact Ribbon Customer Support.

11.2.2 Versioning Information

The module’s versioning information can be viewed by navigating to **Monitoring -> Dashboard -> System Status** from the SBC main screen via the EMA. The System Status window display will include the hardware type and SBC application version, which can be correlated with the module name and software version (respectively) on the module’s validation record.

11.2.3 Software Loading

Procedures for loading a new module image can be found under the appropriate entries on Ribbon’s Upgrading SBC Core Software webpage.

11.2.4 High Availability Configuration

When configuring the module for high availability, the VMs for the active instance and the standby instance must reside on the same physical host for operation in the Approved mode. For connection details, refer to the instructions found under the online document entry “Setting Up Logical Connection Between High Availability Nodes”.

11.2.5 Additional Administrator Policies and Guidance

This section notes additional policies below that must be followed by the CO:

- Once the “fips-140-3 mode” is set to ‘enabled’, it cannot be disabled through the configuration. A fresh software installation is required to set the operational mode back to ‘disabled’.
- To ensure correct functioning and compliance with this Security Policy, module operators must use phones that support TLS v1.2.
- When using local certificate management mode, certificates are first stored in encrypted form on the external workstation prior to being sent to the module via SSH. To ensure that the certificate file can be properly decrypted and installed once sent to the module, module operators must ensure the following:
    - The encryption algorithm used on the external workstation must be 128-bit AES in CBC mode, and the salt length used on the external workstation as input to the PBKDF2 must be 128 bits.

- As the Certificate Load Key used by the module to decrypt the loaded certificate is established using PBKDF2 as specified in *NIST SP 800-132*, the same passphrase that was used on the external workstation must be entered into the module in order to derive the correct key.

11.3 Non-Administrator Guidance

With the exception of the User password, User role operators do not have the ability to configure sensitive information on the module. The User must be diligent to select strong passwords in adherence to the password policies set by the CO from section 4.1 and must not reveal their password to anyone. Additionally, User role operators should be careful to protect any secret or private keys to which they have access.

11.4 Design and Rules

By design, the module follows or enforces the following rules of operation:

- The module provides two distinct operator roles: User and Cryptographic Officer.
- An operator does not have access to any cryptographic services prior to assuming an authorized role.
- The module performs all self-tests without any operator action required.
- The module inhibits data output during key generation, self-tests, zeroization, and error states.
- Status information output by the module does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
- The module does not support a maintenance interface or role.
- The module does not support manual SSP establishment methods.
- The module does not have any proprietary external input/output devices used for entry/output of data.
- The module does not output intermediate key values.
- The module does not provide bypass services or ports/interfaces.

11.5 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of performing the zeroization methods described in section 9.3 above. This will ensure that any SSPs in volatile memory are zeroized.

12. Mitigation of Other Attacks

The module does not claim to mitigate any attacks beyond the FIPS 140-3 Level 1 requirements for this validation. Therefore, per *ISO/IEC 19790:2012* section 7.12, requirements for this section are not applicable.

Appendix A. Acronyms and Abbreviations

Table 32 provides definitions for the acronyms and abbreviations used in this document.

Table 32. Acronyms and Abbreviations

Term	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CCCS	Canadian Centre for Cyber Security
CMVP	Cryptographic Module Validation Program
CO	Cryptographic Officer
CPU	Central Processing Unit
CSP	Critical Security Parameter
CTR	Counter
CVL	Component Validation List
DES	Data Encryption Standard
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMI/EMC	Electromagnetic Interference /Electromagnetic Compatibility
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
GMAC	Galois Message Authentication Code
GPC	General-Purpose Computer
HMAC	(keyed-) Hash Message Authentication Code
KAS	Key Agreement Scheme
KAT	Known Answer Test
KTS	Key Transport Scheme
KW	Key Wrap
KWP	Key Wrap with Padding
NIST	National Institute of Standards and Technology
OS	Operating System
PCT	Pairwise Consistency Test

Term	Definition
PKCS	Public Key Cryptography Standard
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, and Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SP	Special Publication

Prepared by:
Corsec Security, Inc.

12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
Web: www.corsec.com