# Ribbon Communications, Inc.

## SBC 5400 Session Border Controller

Firmware Version: R7.2.1S0

# FIPS 140-2 Non-Proprietary Security Policy

**FIPS Security Level: 2**
**Document Version: 0.2**

**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

# List of Tables

# List of Figures

# 1. Introduction

## 1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the SBC 5400 Session Border Controller from Ribbon Communications, Inc. (Ribbon). This Security Policy describes how the SBC 5400 Session Border Controller meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the U.S. National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 2 FIPS 140-2 validation of the module. The SBC 5400 Session Border Controller is referred to in this document as the SBC 5400 or the module.

## 1.2 References

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

- The Ribbon website (www.ribboncommunications.com) contains information on the full line of products 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.

## 1.3 Document Organization

The Security Policy document is organized into two (2) primary sections. Section 2 provides an overview of the validated module. This includes a general description of the capabilities and the use of cryptography, as well as a presentation of the validation level achieved in each applicable functional area of the FIPS standard. It also provides high-level descriptions of how the module meets FIPS requirements in each functional area. Section 3 documents the guidance needed for the secure use of the module, including initial setup instructions and management methods and policies.

This Security Policy and other validation submission documentation were produced by Corsec Security, Inc. under contract to Ribbon. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Submission Package is proprietary to Ribbon and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Ribbon.

# 2. SBC 5400 Session Border Controller

## 2.1 Overview

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 Controllers (SBC 5400) 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.

Ribbon's SBC 5400 Session Border Controller (see Figure 1 below) is a high-performance air-cooled, 2U, IP encryption appliance that features a unique architecture design that differs from other session border controllers on the market today by aggregating all of the session border functionality – security, encryption, transcoding, call routing, and session management – into a single device, and then distributing those functions to embedded and modular hardware within the device. The SBC 5400 provides secure SIP-based communications with robust security, reduced latency, real-time encryption (VOIP[3] signaling and media traffic), media transcoding, flexible SIP session routing, and policy management.

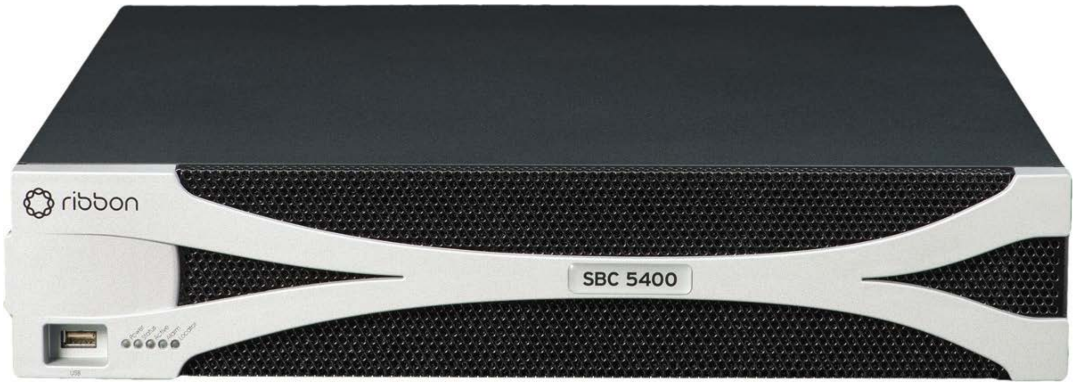

**Figure 1 – SBC 5400 Session Border Controller**

The SBC 5400 is designed to fully address the next-generation need of SIP communications by delivering embedded media transcoding, robust security and advanced call routing in a high-performance, medium form-factor device. The SBC 5400 is designed to accommodate up to 75,000 call sessions. Some of the network and security features provided by the module include:

[1] IP – Internet Protocol

[2] SIP – Session Initiation Protocol

[3] VOIP – Voice Over Internet Protocol

- Session-aware firewall, split DMZ[4], bandwidth & QoS[5] theft protection, topology hiding, DoS[6]/DDoS[7] detection/blocking, rogue RTP[8] protection, IPsec[9] and TLS[10] encryption
- Embedded media transcoding hardware
- H.323 and SIP-I/T interworking
- Stateful call-handling even during overload/attack/outages
- Embedded localized or centralized call-routing options
- Far-end NAT[11] traversal
- TLS, IPsec (IKEv1[12]) for signaling encryption
- Secure RTP/RTCP[13] for media encryption
- Support for large number of protocols including IPv4, IPv6, IPv4/IPv6 interworking, SSH[14], SFTP[15], SNMP[16], HTTPS[17], RTP/RTCP, UDP[18], TCP[19], DNS[20], and ENUM[21]
- Exceptional scalability even under heavy workloads
- Device management using encrypted and authenticated device management messages
- Controlled menu access and comprehensive audit logs
- Integrated Baseband Management Controller (BMC)

The validated module is a solution that delivers end-to-end SIP session control and a networkwide view of SIP traffic and policy management. The module can be deployed as a peering SBC, access SBC, or enterprise SBC.

Figure 2 below illustrates a typical deployment scenario of the SBC 5400.

[4] DMZ – Demilitarized Zone
[5] QoS – Quality of Service
[6] DoS – Denial of Service
[7] DoS/DDoS – Denial-of-Service/Distributed Denial-of-Service
[8] RTP – Real-time Transport Protocol
[9] IPsec – Internet Protocol Secuirty
[10] TLS – Transport Layer Secuirty
[11] NAT – Network Address Translation
[12] IKEv1 – Internet Key Exchange version 1
[13] RTCP – RTP Control Protocol
[14] SSH – Secure Shell
[15] SFTP – SSH File Transport Protocol
[16] SNMP – Simple Network Management Protocol
[17] HTTPS – Hypertext Transfer Protocol Secure
[18] UDP – User Datagram Protocol
[19] TCP – Transmission Control Protocol
[20] DNS – Domain Name System
[21] ENUM – E.164 NUmber Mapping

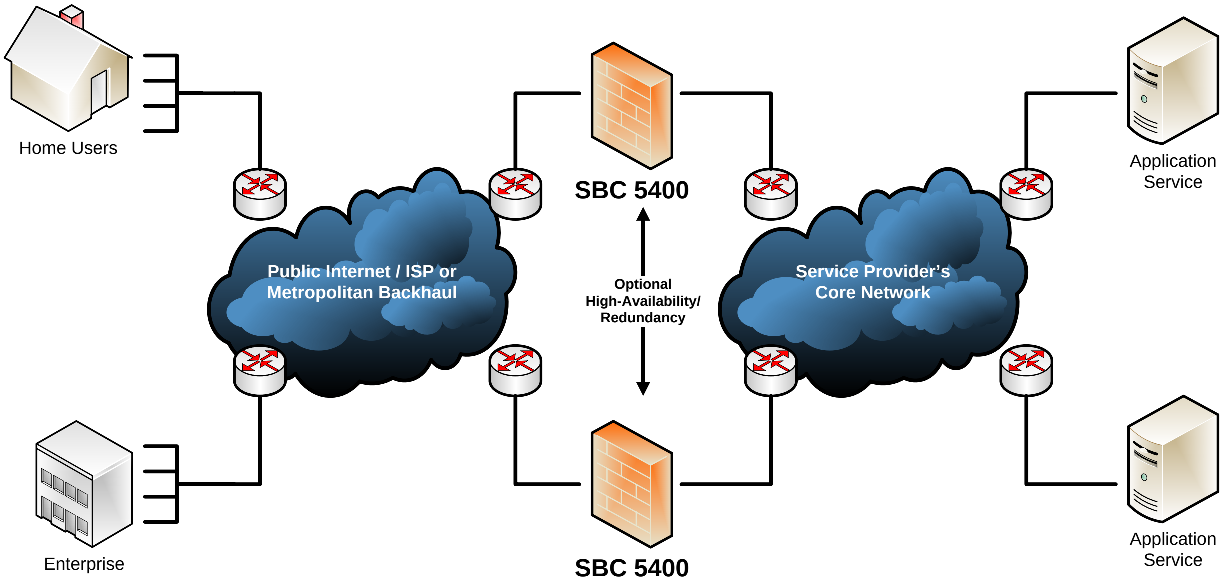
Home Users
Enterprise
Public Internet / ISP or Metropolitan Backhaul
SBC 5400
Optional High-Availability/ Redundancy
SBC 5400
Service Provider's Core Network
Application Service
Application Service

**Figure 2 – A Typical Deployment Scenario of SBC 5400**

Management of the SBC 5400 Session Border Controller is accomplished via:

- 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
- SNMPv3 traps and polling, which are used only for non-security relevant information about the module's state and statistics

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 5400.

The SBC 5400 Session Border Controller is validated at the FIPS 140-2 section levels shown in Table 1 below.

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

| Section | Section Title | Level |
|---|---|---|
| 1 | Cryptographic Module Specification | 2 |
| 2 | Cryptographic Module Ports and Interfaces | 2 |
| 3 | Roles, Services, and Authentication | 2 |
| 4 | Finite State Model | 2 |
| 5 | Physical Security | 2 |

| Section | Section Title | Level |
|---|---|---|
| 6 | Operational Environment | N/A[22] |
| 7 | Cryptographic Key Management | 2 |
| 8 | EMI/EMC[23] | 2 |
| 9 | Self-Tests | 2 |
| 10 | Design Assurance | 2 |
| 11 | Mitigation of Other Attacks | N/A |

## 2.2 Module Specification

The SBC 5400 Session Border Controller is a hardware cryptographic module with a multiple-chip standalone embodiment. The cryptographic module runs Sonus' proprietary ConnexIP operating system (OS), and consists of firmware and hardware components enclosed in a secure, production-grade metal case. The main hardware components consist of integrated circuits, processors, memories, SSD[24], flash, DSP cards, power supplies, fans, and the enclosure containing all of these components. The overall security level of the module is 2. The cryptographic boundary of the SBC 5400 is defined by the SBC 5400 device enclosure, which encompasses all the hardware and firmware components.

### 2.2.1 Excluded Components

BMC functionality is provided by ASPEED Technology's AST2400 Server Management Processor (throughout this document, the AST2400 processor is referred to as "the BMC"). This component is excluded from the security requirements of this standard, along with all supporting chips, circuitry and external ports connected to the BMC. Although it physically resides within the cryptographic boundary, the BMC is a completely independent computing platform, with its own CPU, RAM, flash, ports, and operating system.

Also excluded from the FIPS 140-2 requirements are the Small Form-Factor Pluggable (SFP) transceiver modules that can be connected to the SBC 5400's media and HA ports. These are simply adapters to interface the SBC 5400's ports to either copper-based or fiber-based wiring, depending on the customer need. The SFPs do not provide any cryptographic services, nor do they store or process any critical security parameters. The malfunction of the SFPs cannot cause a breach to the security of the module or the information protected by them.

### 2.2.2 Algorithm Implementations

The SBC 5400 implements cryptographic algorithms, components, and key derivation functions in the following providers:

- Sonus Cryptographic Library 3.1 (Cert. #C868, #A3567)
- Sonus Cryptographic Media Processor 3.1 (Cert. #5676, #2063, #3779, #2845, #4549)
- Sonus SSH KDF Library 3.1 (Cert. #C869)

[22] N/A – Not applicable
[23] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility
[24] SSD – Solid-State Drive

- Sonus TLS KDF Library 3.1 (Cert. #C870)

The Approved algorithms are listed in Table 2 below.

**Table 2 – FIPS-Approved Algorithm Implementations**

| Certificate Number | | Algorithm | Standard | Mode / Method | Key Lengths / Curves / Moduli | Use |
|---|---|---|---|---|---|---|
| Media Processor | Crypto Libraries | | | | | |
| #5676 | - | AES[25] | FIPS PUB 197 | CBC, CTR[26] | 128, 192, 256 | Encryption/decryption |
| | | | NIST SP 800-38D | GCM[27] | 128, 256 | Encryption/decryption |
| - | #C868 | AES | FIPS PUB 197 | CBC, CFB1[28], CFB8, CFB128 | 128, 192, 256 | Encryption/decryption |
| | | | NIST SP 800-38D | GCM | 128, 256 | Encryption/decryption |
| - | Vendor Affirmed | CKG[29] | NIST SP 800-133 | - | - | Symmetric key generation |
| #2063 | - | CVL | NIST SP 800-135rev1 | SRTP | - | Key derivation<br>*No parts of the SRTP protocol, other than the KDF, have been tested by the CAVP and CMVP.* |
| - | #C869 | CVL | NIST SP 800-135rev1 | SSH v2 | - | Key derivation<br>*No parts of the SSH protocol, other than the KDF, have been tested by the CAVP and CMVP.* |
| - | #C870 | CVL | NIST SP 800-135rev1 | TLS v1.2 | - | Key derivation<br>*No parts of the TLS protocol, other than the KDF, have been tested by the CAVP and CMVP.* |
| - | #C868 | DRBG[30] | NIST SP 800-90Arev1 | CTR-based | 128 | Deterministic random bit generation |
| - | #A3567 | DSA | FIPS PUB 186-4 | - | 2048/224, 2048/256, 3072/256 | Key pair generation |
| - | #C868 | ECDSA[31] | FIPS PUB 186-4 | - | B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 | Key pair generation |

[25] AES – Advance Encryption Standard
[26] CTR – Counter
[27] GCM – Galois Counter Mode
[28] CFB – Cipher Feedback
[29] CKG – Cryptographic Key Generation
[30] DBRG – Deterministic Random Bit Generator
[31] ECDSA – Elliptic Curve Digital Signature Algorithm

| Certificate Number | | Algorithm | Standard | Mode / Method | Key Lengths / Curves / Moduli | Use |
|---|---|---|---|---|---|---|
| Media Processor | Crypto Libraries | | | | | |
| | | | | - | B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521 | Key pair verification |
| | | | | SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 | B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 | Digital signature generation |
| | | | | SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 | B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521 | Digital signature verification |
| #3779 | - | HMAC[32] | FIPS PUB 198-1 | SHA-1[33] | - | Message authentication |
| - | #C868 | HMAC | FIPS PUB 198-1 | SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 | - | Message authentication |
| - | #A3567 | KAS | NIST SP 800-56Arev3 | KAS-ECC-SCC with KDF (SRTP, SSH, TLS) | P-224, P-256, P-384, P-521 | Key agreement<br>*Key establishment methodology provides between 112 and 256 bits of encryption strength* |
| - | #A3567 | KAS | NIST SP 800-56Arev3 | KAS-FFC-SCC with KDF (SRTP, SSH, TLS) | 2048/224 (FB), 2048/256 (FC) | Key agreement<br>*Key establishment methodology provides 112 bits of encryption strength* |
| - | #A3567 | KAS-ECC-SSC[34] | NIST SP 800-56Arev3 | ephemeralUnified | P-224, P-256, P-384, P-521 | Shared secret computation |
| - | #A3567 | KAS-FFC-SSC[35] | NIST SP 800-56Arev3 | dhEphem | 2048/224 (FB), 2048/256 (FC) | Shared secret computation |
| - | #C868 | KTS[36] | NIST SP 800-38D | AES GCM | 128, 256 | Key wrap/unwrap[37]<br>*Key establishment methodology provides 128 or 256 bits of encryption strength* |

[32] HMAC – (keyed-) Hashed Message Authentication Code

[33] SHA – Secure Hash Algorithm

[34] KAS-FFC-SSC – Key Agreement Scheme – Elliptic Curve Cryptography - Shared Secret Computation

[35] KAS-FFC-SSC – Key Agreement Scheme - Finite Field Cryptography - Shared Secret Computation

[36] KTS – Key Transport Scheme

[37] Per FIPS 140-2 Implementation Guidance D.9, AES GCM is an Approved method for key transport.

| Certificate Number | | Algorithm | Standard | Mode / Method | Key Lengths / Curves / Moduli | Use |
|---|---|---|---|---|---|---|
| Media Processor | Crypto Libraries | | | | | |
| | | | FIPS PUB 197<br>FIPS PUB 198-1 | AES and HMAC | 128, 192, 256 | Key wrap/unwrap[38]<br>*Key establishment methodology provides between 128 and 256 bits of encryption strength* |
| - | Vendor Affirmed | PBKDF2[39] | NIST SP 800-132 | Option 1 | - | Key derivation |
| - | #C868 | RSA[40] | FIPS PUB 186-4 | ANSI[41] X9.31 | 2048 | Key pair generation |
| | | | | PKCS#1 v1.5 | 2048 | Digital signature generation |
| | | | | | 1024, 2048 | Digital signature verification |
| #4549 | | SHS[42] | FIPS PUB 180-4 | SHA-1 | - | Message digest |
| - | #C868 | SHS | FIPS PUB 180-4 | SHA2-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 | - | Message digest |
| #2845 | - | Triple-DES[43] | NIST SP 800-67rev2 | TCBC | Keying option 1 | Encryption/decryption |
| - | #C868 | Triple-DES | NIST SP 800-67rev2 | TCBC | Keying option 1 | Encryption/decryption |

The vendor affirms the following cryptographic security methods:

- Crytographic key generation – As per *NIST SP 800-133*, the module uses its FIPS-Approved counter-based DRBG to generate cryptographic keys. The resulting symmetric key or generated seed is an unmodified output from the DRBG. The module's DRBG is seeded via `/dev/random`, a non-deterministic random number generator (NDRNG) internal to the module.

- Password-based key derivation – As per *NIST SP 800-132*, the module uses PBKDF2 option 1 to derive the Certificate Load Key. This function takes an input salt that is 128 bits in length with a passphrase containing at least eight characters (following the password complexity requirements in section 3.4 below) and produces a random value of 128 bits (when producing keys for AES) or 168 bits (when producing keys for Triple DES). In addition, the function has an iteration count of 2048. The underlying pseudorandom function used in this derivation is SHA-1.

The module also implements 4096-bit RSA signature generation. Per FIPS 140-2 Implementation Guidance A.14, "when performing an RSA signature generation, a module may use any modulus size greater than or equal to 2048 bits".

[38] Per FIPS 140-2 Implementation Guidance D.9, AES (any approved mode) with HMAC is an Approved method for key transport.
[39] PBKDF2 – Password-Based Key Derivation Function 2
[40] RSA – Rivest Shamir Adleman
[41] ANSI – American National Standards Institute
[42] SHS – Secure Hash Standard
[43] DES – Data Encryption Standard

The module implements the non-Approved but allowed algorithms shown in Table 3.

**Table 3 – Allowed Algorithms**

| Algorithm | Caveat | Use |
|---|---|---|
| NDRNG[44] | - | Seeding for the FIPS-Approved DRBG |
| RSA | key establishment methodology provides 112 bits of encryption strength | Key transport |

The module implements the following non-Approved algorithms (use of these algorithms is restricted to the module's non-Approved mode of operation):

- IKE v1/v2 KDF (non-compliant)
- MD5 (when used for data authentication in IPsec sessions)

### 2.2.3 Modes of Operation

The module supports two modes of operation: Approved and Non-approved. The module will be in FIPS-Approved mode when all power up self-tests have completed successfully, and only Approved algorithms are invoked. See Table 2 and Table 3 above for a list of the Approved and allowed algorithms (respectively).

When in the operational state, the module can alternate service-by-service between Approved and non-Approved modes of operation. The module will switch to the non-Approved mode upon execution of a non-Approved service. The module will switch back to the Approved mode upon execution of an Approved service. No keys or CSPs are shared between modes

The module supports the Crypto Officer and User roles while in the non-Approved mode of operation.

Table 4 below lists the services available in the non-Approved mode of operation.

**Table 4 – Non-Approved Services**

| Service | Operator | | Description |
|---|---|---|---|
| | CO | User | |
| Establish IPsec Session | ✓ | ✓ | Establish  remote session using IPsec protocol |

[44] NDRNG – Non Deterministic Random Number Generator

## 2.3 Module Ports and Interfaces

The module's design separates the physical ports and interfaces into four logically distinct and isolated categories. They are:

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

Data input/output are the packets utilizing the services provided by the module. These packets enter and exit the module through the Ethernet media, management, and HA[45] interfaces. Control input consists of configuration or administration data entered into the module through the Command Line Interface (CLI) and Web GUI over Ethernet management interfaces and HA ports. Status output consists of the status relayed over the Ethernet management interfaces, HA ports, and also displayed via LEDs[46] and through the log information accessible over Ethernet management ports.

### 2.3.1 Front Interfaces

The physical LEDs and ports/interfaces found on the front bezel of the SBC 5400 Session Border Controller are shown in Figure 3 below.

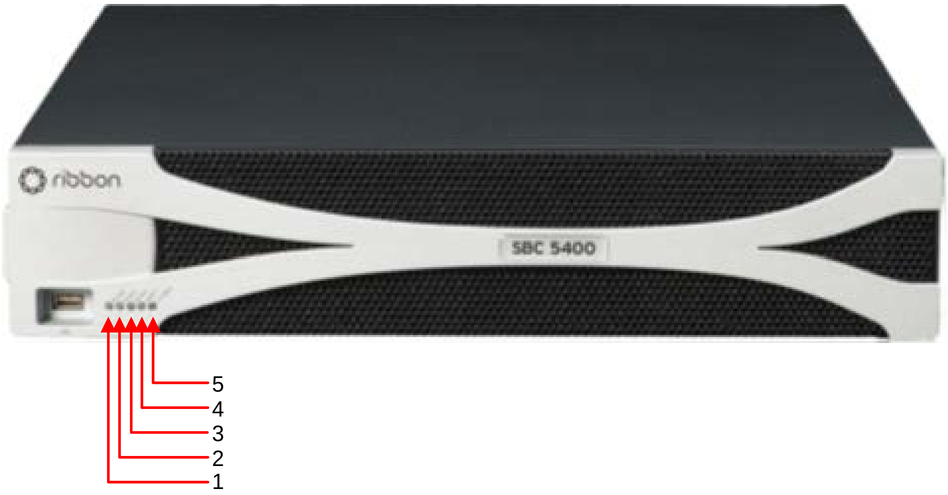

**Figure 3 – Front Bezel Ports/Interfaces and LEDs**

Table 5 provides the mapping from the physical interfaces on the front bezel to logical interfaces as defined by FIPS 140-2.

[45] HA – High Availability

[46] LED – Light Emitting Diode

**Table 5 – FIPS 140-2 Logical Interface Mappings (Front Bezel)**

| Physical Port/Interface | Color | Description | FIPS 140-2 Logical Interface |
|---|---|---|---|
| (1) Front Power LED | Amber/Green | Power status indicator:<br>• GREEN: all power on<br>• OFF: BMC power on or chassis power off | • Status out |
| (2) Front Status LED | Amber/Green | Module status indicator:<br>• GREEN: application is running<br>• AMBER: application startup has not completed, or the application has been shutdown<br>• OFF: application is not running | • Status out |
| (3) Front Active LED | Amber/Green | Module Redundancy State indicator:<br>• GREEN: active and protected<br>• AMBER: active but unprotected<br>• OFF: not active | • Status out |
| (4) Front Alarm LED | Amber/Red | Module critical/major failure indicator:<br>• RED: critical alarm<br>• AMBER: major alarm<br>• OFF: no alarm conditions | • Status out |
| (5) Front Locator LED | White | Module identifier indicator (ON or BLINKING, depending on the "ipmitool chassis identify" command) | • Status out |

***Note**: The module also includes a front-facing USB[47] port that allows for the connection of external devices for firmware image downloads. Use of this USB port for any purpose is prohibited while the module is operating in its FIPS-Approved mode.*

The physical LEDs and ports/interfaces found behind the front bezel of the SBC 5400 Session Border Controller are shown in Figure 4 below.

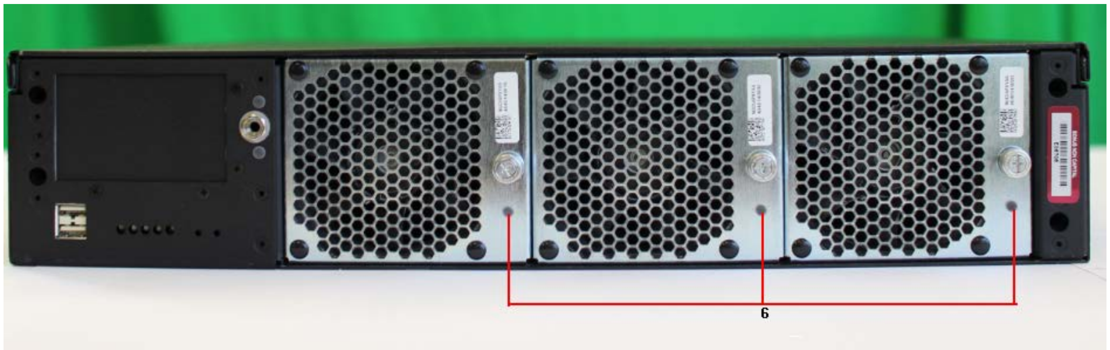

**Figure 4 – Front Panel Ports/Interfaces and LEDs**

[47] USB – Universal Serial Bus

Table 6 provides the mapping from the physical interfaces behind the front bezel to logical interfaces as defined by FIPS 140-2.

**Table 6 – FIPS 140-2 Logical Interface Mappings (Front Panel)**

| Physical Port/Interface | Color | Description | FIPS 140-2 Logical Interface |
|---|---|---|---|
| (6) Fan Control LED | Green | Fan module indicator (1 per fan):<br>• GREEN: fan module is working<br>• OFF: fan module is not working | • Status out |

Please note that the front bezel LEDs also show on the module's front panel when the bezel is not mounted.

## 2.3.2 Rear Interfaces

The physical ports and interfaces found on the rear panel of the SBC 5400 Session Border Controller are shown in Figure 5 below.

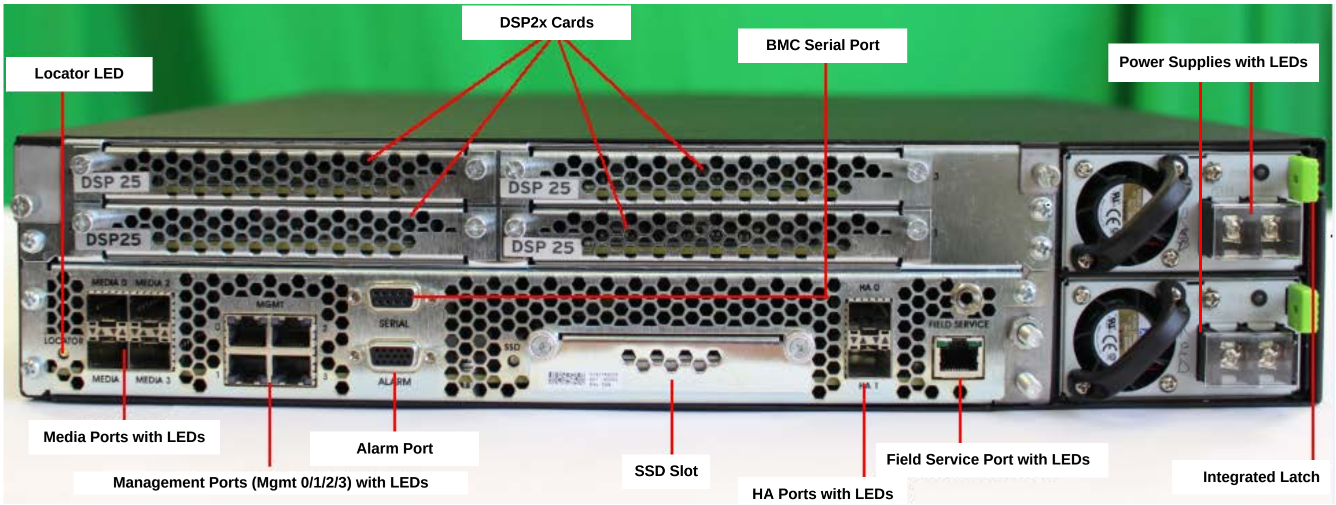

**Figure 5 – Rear Panel Ports/Interfaces and LEDs**

Table 7 provides the mapping from the physical ports/interfaces on the rear panel to logical interfaces as defined by FIPS 140-2.

Table 7 – FIPS 140-2 Logical Interface Mappings (Rear Panel)

| Physical Port/Interface | Quantity | Description | FIPS 140-2 Logical Interface |
|---|---|---|---|
| Management Ports | 4 x 100 Mbps[48] | Copper RJ-45 Ethernet ports which process management traffic and perform protocol termination | • Data in<br>• Data out<br>• Control in<br>• Status out |
| Management Port LEDs | 2 per port | Indicator of management port link and activity status:<br>• The Link LED is solid green when the link is up.<br>• The Activity LED is Flashing Green when there is activity present on the port. | • Status out |
| Locator LED | 1 | Module identifier indicator | • Status out |
| Media Ports | 4 x 1 Gbps[49]<br>(two ports are 10 Gbps-capable) | Copper SFP or fiber SFP Ethernet ports for media and signaling traffic | • Data in<br>• Data out |
| Media Port LEDs | 2 per port | Indicator of media port link and activity status:<br>• The Link LED is solid green when the link is up.<br>• The Activity LED is Flashing Green when there is activity present on the port. | • Status out |
| SSD Slot | 1 | Solid state drive slot | • Data in |
| High Availability Ports | 2 x 1 Gbps | Copper SFP or fiber Ethernet SFP ports for redundancy synchronization traffic. | • Data in<br>• Data out<br>• Control in<br>• Status out |
| High Availability Port LEDs | 2 x 1 Gbps | Indicator of HA port link and activity status:<br>• The Link LED is solid green when the link is up.<br>• The Activity LED is Flashing Green when there is activity present on the port. | • Status out |
| BMC Serial Port | 1 | An RS-232 port used by BMC | N/A |
| Field Service Ethernet Port | 1 x 100 Mbps | An Ethernet port for servicing in the field | N/A |
| Field Service Ethernet Port LEDs | 2 per port | Indicator of field service port link and activity status:<br>• The Link LED is solid green when the link is up.<br>• The Activity LED is Flashing Green when there is activity present on the port. | • Status out |

[48] Mbps – Megabits per second
[49] Gbps – Gigabits per second

| Physical Port/Interface | Quantity | Description | FIPS 140-2 Logical Interface |
|---|---|---|---|
| Power LED | 1 per power supply | Indicator of power supply status:<br>• OFF: No power input to the supply<br>• AMBER: Power supply fault<br>• BLINKING AMBER: Main system power output fault<br>• BLINKING GREEN: Standby power to BMC is ON but main system power is not enabled<br>• GREEN: Both standby power to BMC and main system power are on and OK. | • Status out |

***Note****: Each module also includes a back panel alarm port. This port is not operational, and thus provides no facility for input or output.*

## 2.4 Roles, Services, and Authentication

The sections below describe the module's roles and services, and define any authentication methods employed.

### 2.4.1 Authorized Roles

As required by FIPS 140-2, the module supports two 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 CSPs. The CO is also responsible for the management of all keys and CSPs, including their zeroization. Lastly, the CO is the only operator that can configure the module into FIPS-Approved mode of operation. The CO also has access to all User services.

- 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.

### 2.4.2 Operator Services

Descriptions of the services available to the Crypto Officer role and User role are provided in the Table 8 below. The keys and CSPs listed in Table 8 indicate the type of access required using the following notation:

- R – Read: The CSP is read.
- W – Write: The CSP is established, generated, modified, or zeroized.
- X – Execute: The CSP is used within an Approved or Allowed security function or authentication mechanism.

**Table 8 – Authorized Operator Services**

| Service | Operator | | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|---|---|
| | CO | User | | | | |
| Commission the module | ✓ | | Commission the module by following the Security Policy guidelines | None | None | None |
| Manage SBC license | ✓ | | Installs the license to enable SBC features; delete or update license; view current license status | Command | Status output | None |
| Configure the SBC system | ✓ | | Define network interfaces and settings; set protocols; configure authentication information; define policies and profiles | Command and parameter | Command response/ Status output | None |
| Configure routing policy and control services | ✓ | | Configure IP network parameters and profiles for signaling, media, call routing, call services, zone, IP ACL[50] rules, NTP[51] and DNS[52] servers | Command and parameters | Command response/ Status output | None |
| Configure Call Data Record (CDR) | ✓ | | Configure log file behavior | Command and parameters | Command response/ Status output | None |
| Manage users | ✓ | | Create, edit and delete users; define user accounts and assign permissions. | Command and parameters | Command response/ Status output | Crypto Officer Password – R/W/X<br>User Password – R/W/X |
| Manage user sessions | ✓ | | Terminate User sessions | Command and parameters | Command response/ Status output | TLS Session Key – W |
| Change password | ✓ | ✓ | Modify existing login passwords | Command and parameters | Command response/ Status output | Crypto Officer Password – R/W<br>User Password – R/W |

[50] ACL – Access Control List
[51] NTP – Network Time Protocol
[52] DNS – Domain Name System

| Service | Operator | | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|---|---|
| | CO | User | | | | |
| Load certificate | ✓ | | Load new certificates | Command | Command response/ Status output | Certificate Load Key – W/X<br>CA[53] Public Key – R/W<br>TLS Private Key – R/W<br>TLS Public Key – R/W<br>TLS Peer Public Key – R/W<br>SSH Peer Public Key – R/W |
| Run script | ✓ | | Run a script file (a text file containing a list of CLI commands to execute in sequence) | Command | Command response/ Status output | None (service may potentially access CSPs indirectly via scripted CLI commands) |
| Perform self tests | ✓ | | Perform on-demand Self-Tests | Command | Command response/ Status output | All ephermeral keys and CSPs – W |
| Perform network diagnostics | ✓ | ✓ | Monitor connections (e.g. ping) | Command | Command response/ Status output | None |
| Show status | ✓ | ✓ | Show the system status, Ethernet status, FIPS Approved mode, alarms, system identification and configuration settings of the module | Command | Command response/ Status output | None |
| Manage event logs | ✓ | | Set cryptographic protections of log files; generate/import log hash signing keys; sign/verify log hashes; retrieve key data; view event status messages | Command | Command response/ Status output | Default Log Signing Key – R/W/X<br>Default Log Verify Key – R/W/X<br>User Log Signing Key – R/X |
| Zeroize keys | ✓ | | Zeroize all keys and CSPs | Command | Command response/ Status output | All ephemeral and persistent keys and CSPs – W |
| Upgrade firmware | ✓ | | Load new firmware and performs an integrity test using an RSA digital signature | Command | Command response/ Status output | Firmware Load Authentication Key – R/X |
| Perform keying of CDB[54] Key | ✓ | | Generate CDB key | Command and parameters | Command response/ Status output | CDB key – W/X |
| Reboot/Reset | ✓ | | Reboot or reset the module | Command | Command response/ Status output | CSPs stored in SDRAM – W |

[53] CA – Certificate Authority
[54] CDB – Configuration Database

| Service | Operator | | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|---|---|
| | CO | User | | | | |
| Establish TLS session | ✓ | ✓ | Establish web session using TLS protocol | Command | Command response/ Status output | AES-GCM IV – R/W/X<br>Diffie-Hellman Public Key – R/X<br>Diffie-Hellman Private Key – X<br>ECDH Public Component – R/X<br>ECDH Private Component – X<br>TLS Private Key – W/X<br>TLS Public Key – W/X<br>TLS Peer Public Key – R/X<br>TLS Pre-Master Secret – W/X<br>TLS Master Secret – W/X<br>TLS Session Key – R/W/X<br>TLS Authentication Key – W/X |
| Establish SSH session | ✓ | ✓ | Establish remote session using SSH protocol | Command | Command response/ Status output | Diffie-Hellman Public Key – R/X<br>Diffie-Hellman Private Key – X<br>ECDH Public Component – R/X<br>ECDH Private Component – X<br>SSH Private Key – W/X<br>SSH Public Key – W/X<br>SSH Peer Public Key – R/X<br>SSH Shared Secret – W/X<br>SSH Session Key – R/W/X<br>SSH Authentication Key – W/X |
| Establish SRTP session | ✓ | ✓ | Establish a SIP/TLS session using SRTP protocol | Command | Command response/ Status output | AES-GCM IV – R/W/X<br>SRTP Master Key – R/X<br>SRTP Session Key – W/X<br>SRTP Authentication Key – W/X |
| Negotiate SFTP session | ✓ | ✓ | Establish an SFTP session | Command | Command response/ Status output | SFTP Private Key – W/X<br>SFTP Public Key – W<br>SFTP Peer Public Key – R/X |
| SNMPv3 traps | | ✓ | Provides system condition information | None | Status output | SNMPv3 Session Key – R/W/X<br>SNMPv3 Authentication Key – R/W/X |
| Encryption/ decryption service | ✓ | ✓ | Encrypt or decrypt user data, keys, or management traffic | Command and parameters | Command response | TLS Session Key – X<br>SSH Session key – X |

All services listed above require the operator to assume a role, and the module authenticates the role before providing any of these services.

### 2.4.3 Additional Services

The module provides a limited number of services for which the operator is not required  to assume an authorized role. Table 9 lists the services for which the operator is not required to assume an authorized role. None of the services listed in the table disclose cryptographic keys and CSPs or otherwise affect the security of the module.

**Table 9 – Additional Services**

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| Zeroize | Zeroize keys and CSPs | Power cycling using power connectors | Status output | All ephemeral keys and CSPs – W |
| Perform on-demand self-tests | Perform power-up self-tests on demand | Cycle power using power connectors | Status output | All ephemeral keys and CSPs – W |
| Authenticate | Used for operator logins to the module | Command | Status output | Crypto Officer Password – X<br>User Password – X<br>RADIUS Shared Secret – W/X<br>TLS Public Key – X |

### 2.4.4 Authentication

The module supports role-based authentication and multiple concurrent operators. Operator authentication is managed either from a local database or a configured remote RADIUS server. Upon initial module configuration, local authentication is enabled by default. If both methods are enabled, external (RADIUS) authentication takes priority and is attempted first. If authentication fails, the module attempts local authentication. The login attempt is rejected if both attempts fail.

All module operators authenticate using a username and password. The module also supports RSA digital certificate authentication of operators during Web GUI/HTTPS (TLS) access. Table 10 lists the authentication mechanisms used by the module. The strength calculation below provides minimum strength based on password policy described in Section 3.4.

**Table 10 – Authentication Mechanism Used by the Module**

| Authentication Type | Strength |
|---|---|
| Password | The minimum length of the password is eight characters, with 95 different case-sensitive alphanumeric characters and symbols possible for usage. The chance of a random attempt falsely succeeding is:<br>$=1/95^8$<br>which is less than 1/1,000,000 as required by FIPS 140-2.<br>The fastest network connection over Ethernet Interface supported by the module is 100 Mbps. Hence, at most $(10 \times 10^7 \times 60) = (6 \times 10^9)$ = 6,000,000,000 bits of data can be transmitted in one minute. The probability that a random attempt will succeed or a false acceptance will occur in one minute is:<br>$=((6 \times 10^9$ bits per minute / 64 bits per password) / $95^8$ possible passwords<br>=93,750,000 passwords per minute / $95^8$ possible passwords<br>$=93{,}750{,}000 / 95^8$<br>which is less than 1/100,000 as required by FIPS 140-2. |
| Public Key Certificates | The module supports RSA digital certificate authentication of users during Web GUI/HTTPS (TLS) access. Using conservative estimates and equating a 2048-bit RSA key to a 112-bit symmetric key, the probability for a random attempt to succeed is:<br>$=1/2^{112}$<br>which is less than 1/1,000,000 as required by FIPS 140-2.<br>The fastest network connection over the Media ports supported by the module is 10 Gbps. Hence, at most $(10 \times 10^9 \times 60) = (6 \times 10^{11})$ = 600,000,000,000 bits of data can be transmitted in one minute. The probability that a random attempt will succeed or a false acceptance will occur in one minute is:<br>$=(6 \times 10^{11}$ bits per minute / 112 bits per key) / $2^{112}$ possible keys<br>$=5{,}357{,}142{,}857 / 2^{112}$<br>which is less than 1/100,000 as required by FIPS 140-2. |

The visual feedback of authentication data is obscured during an operator’s entry of authentication credentials. The module provides feedback by displaying a “rounded dot” (●) symbol when an operator is entering his password over EMA login, while no feedback is provided for CLI login.

The module provides the ability for an operator to change roles. In order to change roles, an operator is required to first log out and then re-authenticate with an account with appropriate permissions for the desired role.

The module does not allow the disclosure, modification, or substitution of authentication data to unauthorized operators. The authenticated CO can modify their own authentication credentials as well as the credentials of the Users, while the Users have the ability to modify their own authentication data only.

## 2.5 Physical Security

The SBC 5400 is a multi-chip standalone cryptographic module. All CSPs are stored and protected within the module’s production-grade enclosure. All of the components within the module are production grade with standard passivation.

The module's chassis is opaque within the visible spectrum. There are a limited set of ventilation holes provided in the case's front bezel and rear panel that obscure visual access to the module's internal components. Tamper-evident labels are applied to the chassis to provide physical evidence of attempts to remove protected components. The placement of the tamper-evident labels can be found in section 3.1.2 of this document.

## 2.6 Operational Environment

The module employs an Intel Xeon (Ivy Bridge) processor running Ribbon's proprietary ConnexIP OS. The media procesor is a Cavium OCTEON II CN6880. This operational environment does not provide a general-purpose OS to the operator. The operational environment is not modifiable by the operator, and only the module's signed image can be executed. All firmware upgrades are digitally-signed, and a conditional self-test (RSA signature verification) is performed during each upgrade. If the signature test fails, the new firmware is ignored and the current firmware remains loaded.

**NOTE**: Only FIPS-validated firmware may be loaded to maintain the module's validation.

## 2.7 Cryptographic Key Management

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

- Local – RSA public/private key pairs and Certificate Signing Requests (CSRs) for the SBC 5400 are generated on an external workstation. Each CSR is signed with workstation's private key and then submitted to a Certificate Authority (CA). The workstation receives the issued certificate back from the CA, then stores the key pair and certificate in a PKCS #12-formatted file. This certificate file is then encrypted (using 128-bit AES or Triple-DES in CBC mode) and sent to the SBC 5400 via SSH for installation.

- Local-Internal – The SBC 5400 generates its RSA key pairs and Certificate Signing Requests (CSR) internally. The certificate request is signed with SBC 5400's private key and then sent to a CA. The issued certificate is received back from the CA and then installed on the SBC 5400.

- 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.

The module supports the secret/private keys, key components, and CSPs described in Table 11, and the public keys and key components in Table 12 below.

**Table 11 – Secret/Private Keys, Key Components, and CSPs**

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| Config Database (CDB) Key | Triple-DES 168-bit key | Generated internally via FIPS-Approved DRBG | Never exits the module | Plaintext on SSD | When re-keyed over CLI or EMA;<br>When appliance is re-imaged;<br>Upon command via CLI or EMA | Encryption/decryption of RSA and ECDSA private keys and preshared secrets for RADIUS in CDB |
| ECDH Private Component | Private component of ECDH protocol | Generated internally | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Input to SSH and TLS KDFs for ECDH shared secret computation |
| Diffie-Hellman Private Key | 2048-bit DH key | Generated internally | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Input to SSH and TLS KDFs for DH shared secret computation |
| SSH Private Key | 2048-bit RSA key | Generated internally via FIPS-Approved DRBG | Never exits the module | Encrypted in the CDB on SSD | Upon command via CLI or EMA | Authentication during SSH session negotiation |
| SSH Shared Secret | Shared secret | Derived internally via DH/ECDH shared secret computation | Never exits the module | Not persistently stored | Upon module reboot; Upon session termination | Derivation of the SSH Session Key and SSH Authentication Key |
| SSH Session Key | 128/192/256 AES-CBC, 128/192/256 AES-CTR, or 168-bit Triple-DES CBC key | Derived internally via SSH KDF | Never exits the module | Not persistently stored | Upon module reboot; Upon session termination | Encryption and decryption of SSH session packets |
| SSH Authentication Key | 160-bit (minimum) HMAC key | Derived internally via SSH KDF | Never exits the module | Not persistently stored | Upon module reboot; Upon session termination | Authentication of SSH session packets |

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| TLS Private Key | [for authentication using RSA certificates] 2048-bit RSA private key<br>[for authentication using ECDSA certificates] All NIST-recommended ECDSA curves | [for local-internal certificates] Generated internally via FIPS-Approved DRBG<br>[for local certificates] Generated externally, imported via PKCS #12 file format in encrypted form | Never exits the module | Encrypted in the CDB on SSD | Upon command via CLI or EMA | Authentication during TLS key negotiation |
| TLS Pre-Master Secret | [for RSA cipher suites] 384-bit random value<br>[for DH/ECDH cipher suites] DH/ECDH shared secret | [for RSA cipher suites and module acting as client] Generated internally via FIPS-Approved DRBG<br>[for RSA cipher suites and module acting as server] Generated externally, imported in encrypted form via RSA key transport<br>[for DH/ECDH cipher suites] Derived internally via DH/ECDH shared secret computation | [for RSA cipher suites and module acting as client] Exits the module in encrypted form via RSA key transport<br>[for RSA cipher suites and module acting as server] Never exits the module<br>[for DH/ECDH cipher suites] Never exits the module | Not persistently stored | Upon module reboot;<br>Upon completion of TLS Master Secret computation | Derivation of the TLS Master Secret |
| TLS Master Secret | 384-bit shared secret | Derived internally using the TLS Pre-Master Secret via TLS KDF | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Derivation of the TLS Session Key and TLS Authentication Key |
| TLS Session Key | 128/256-bit AES key or 128/256-bit AES-GCM key | Derived internally using the TLS Master Secret via TLS KDF | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Encryption and decryption of TLS session packets |

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| TLS Authentication Key | 160-bit (minimum) HMAC key | Derived internally using the TLS Master Secret via the TLS KDF | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Authentication of TLS session packets |
| SRTP Master Key | 128/192/256-bit shared secret | Generated externally, imported in encrypted form via a secure SIP/TLS session | Exits in encrypted form | Not persistently stored | Upon module reboot;<br>Upon session termination | Peer Authentication, Session and Authentication keys derivation for SRTP session |
| SRTP Session Key | 128/192/256-bit AES-CTR or 128/256-bit AES GCM key | Generated internally using SRTP Master Key | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Encryption or decryption during SRTP session |
| SRTP Authentication Key | 160-bit HMAC key | Generated internally using SRTP Master Key | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Authentication of SRTP session packets |
| RADIUS Shared Secret | Shared secret (alpha-numeric string) | Entered electronically by Crypto Officer | Never exits the module | Encrypted in the CDB on SSD | Upon command via CLI or EMA | Peer authentication of RADIUS messages |
| Default Log Signing Key | 4096-bit RSA private key | Generated internally | Never exits the module | Encrypted in the CDB on SSD | Upon generation of a new key pair | Generation of signatures of stored audit log and security event log hashes |
| User Log Signing Key | 2048-bit (minimum) RSA private key | Generated externally, entered in encrypted form via secure TLS session | Never exits the module | Encrypted in the CDB on SSD | Upon command via CLI | Generation of signatures of stored audit log and security event log hashes |
| DRBG Seed | 256-bit value | Generated internally using entropy input string | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Seed value generated by the DRBG |

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| DRBG Entropy Input String[55] | 512-bit value | Generated internally from various module resources by the NDRNG | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Entropy material for DRBG seed generation |
| DRBG Key Value | Internal DRBG state value | Generated internally | Never exits the module | Not persistently stored | Upon module reboot | Generation of random number |
| DRBG 'V' Value | Internal DRBG state value | Generated internally | Never exits the module | Plaintext in RAM | Upon module reboot | Generation of random number |
| SFTP Private Key | 2048-bit RSA key | Generated internally via FIPS-Approved DRBG | Never exit the module | Encrypted (for the certificates) in the CDB on SSD<br>Plaintext (for SSH) outside CDB on SSD | Upon command via CLI or EMA | Used for SFTP key negotiation |
| SNMPv3 Session Key | 128-bit AES-CFB or 168-bit Triple-DES key | Generated externally, imported in encrypted form via a secure TLS or SSH session | Exits in encrypted form (via TLS session) within configuration data when performing configuration backup | Plaintext in the CDB on SSD | Upon command via CLI or EMA | Encrypting SNMPv3 packets |
| SNMPv3 Authentication Key | 160-bit HMAC key | Generated externally, imported in encrypted form via a secure TLS or SSH session | Exits in encrypted form (over TLS session) within configuration data when performing configuration backup | Plaintext in the CDB on SSD | Upon command via CLI or EMA | Authenticating SNMPv3 packets |

[55] With a *min-entropy* lower bound value of 0.748390, the module's entropy scheme provides more than 256 bits of entropy per call, which is sufficient to support the apparent key strength of the generated keys.

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| Crypto Officer Password | Alphanumeric string (minimum of eight characters) | [for default password] Generated externally and embedded in release image<br>[for new password] Generated externally and entered into module via a console port or over SSH | [for default password] Provided to the CO role operator over CLI/EMA via encrypted session<br>[for new password] Never exits the module | [for default password] Hashed[56] in the CDB on SSD<br>[for new password] Hashed in the CDB on SSD | [for default password] When appliance is re-imaged<br>[for new password] When an all-zero password value is entered into module using the EMA or CLI | Authenticating the Crypto Officer to the module |
| User Password | Alphanumeric string (minimum of eight characters) | [for default password] Generated internally using DRBG<br>[for new password] Generated externally and entered into module via a console port or over SSH | [for default password] Provided to the User role operator over CLI/EMA via encrypted session<br>[for new password] Never exits the module | Hashed form on SSD | When an all-zero password value is entered into module using the EMA or CLI | Authenticating the User to the module |
| Certificate Load Key | 128/256-bit AES or 168-bit Triple-DES key | Derived internally via PBKDF2 (option 1)[57] | Never exits the module | Not persistently stored | Upon module reboot | Decrypting PKCS #12 certificate files when imported from an external workstation |
| AES GCM IV[58] | 96-bit IV | Generated internally via FIPS-Approved DRBG | Never exits the module | Not persistently stored | Upon module reboot | IV input to AES-GCM function |

[56] CO and User passwords are hashed using SHA-512 and stored on the SSD. They are temporarily loaded into the memory in hashed form for comparison during a login.

[57] Keys derived from the PBKDF2 function shall only be used for storage applications.

[58] IV – Initialization Vector

**Table 12 – Public Keys and Key Components**

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| CA Public Key | 2048-bit RSA key | Generated externally, imported in DER[59] file format | Never exits the module | Not persistently stored | Upon module reboot | Verification of Certificate Authority signatures |
| ECDH Public Component | Public component of ECDH protocol | [for the module] Generated internally<br>[for a peer] Generated externally, entered into the module (in certificate form) in plaintext | [for the module] Exits the module in plaintext form<br>[for a peer] Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Input to SSH and TLS KDFs for ECDH shared secret computation |
| Diffie-Hellman Public Key | 2048-bit DH key | [for the module] Generated internally<br>[for a peer] Generated externally, entered into the module (in certificate form) in plaintext | [for the module] Exits the module in plaintext form<br>[for a peer] Never exits the module | Not persistently stored | Upon module reboot;<br>Upon session termination | Input to SSH and TLS KDFs for DH shared secret computation |
| SSH Public Key | 1024/2048-bit RSA key | Generated internally via FIPS-Approved DRBG | Exits the module in plaintext form | Plaintext in the CDB on SSD | Upon command via CLI or EMA | Exported to peer for authentication during SSH session negotiation |

[59] DER – Distinguished Encoding Rules

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| SSH Peer Public Key | 1024/2048-bit key | Imported in plaintext | Never exits the module | Plaintext in the CDB on SSD | Upon command via CLI or EMA | Imported from peer for authentication during session negotiation<br>**1024-bit key is used for signature verification only** |
| TLS Public Key | [for authentication using RSA certificates] 2048-bit RSA public key<br>[for authentication using ECDSA certificates] All NIST-recommended ECDSA curves | [for local-internal certificates] Generated internally via FIPS-Approved DRBG<br>[for local certificates] Generated externally, imported via PKCS #12 file format in encrypted form | Exits the module via digital certificate in plaintext form | Plaintext in the CDB on SSD | Upon command via CLI or EMA | Authentication during TLS key negotiation |
| TLS Peer Public Key | [for authentication using RSA certificates] 2048-bit RSA public key<br>[for authentication using ECDSA certificates] All NIST-recommended curves | Generated externally, imported in certificate form in plaintext | Never exits the module | Plaintext in the CDB on SSD | Upon command via CLI or EMA | Certificate-based authentication during TLS key negotiation |
| Default Log Verify Key | 4096-bit RSA public key | Generated internally | Never exits the module | Not persistently stored | Upon module reboot;<br>Upon generation of a new key pair | Verification of signatures for stored audit log and security event log hashes |

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| SFTP Public Key | 1024/2048-bit RSA key | Generated internally via FIPS-Approved DRBG | Exits the module in plaintext form | Plaintext in the CDB on SSD | Upon command via CLI or EMA | Used for SFTP key negotiation<br>**1024-bit key is used for signature verification only** |
| SFTP Peer Public Key | 1024/2048-bit RSA key | Generated externally, entered into the module in plaintext form | Never exits the module | Plaintext in the CDB on SSD | Upon command via CLI or EMA | Used for SFTP key negotiation<br>**1024-bit key is used for signature verification only** |
| Firmware Load Authentication Key | 2048-bit key RSA public key | Generated externally and embedded in release image | Never exits the module | Stored in flash memory | The flash location is write-protected in hardware at the factory (i.e., not writeable by end user) and is not zeroized. | Verifying the RSA signature of the digest of a new firmware load package |

The AES-GCM IV[60] is used in the following protocols:

- For TLS, the AES-GCM IV is internally constructed deterministically as specified in *RFC 5288* and section 8.2.1 of *NIST SP 800-38D*. When the nonce_explicit part of the IV exhausts the maximum number of possible values for a given session key, the module will trigger a handshake to establish a new encryption key.

  The module's IV generation is in compliance with the TLSv1.2 specification and only for use within the TLSv1.2 protocol. The module supports acceptable AES-GCM cipher suites specified in section 3.3.1 of *NIST SP 800-52rev2*.

- For SSH, the AES-GCM IV is internally constructed deterministically as specified in *RFC 5647*. When the invocation counter part of the IV exhausts the maximum number of possible values for a given session key, the module will trigger a handshake to establish a new encryption key.

  The module's IV generation is in compliance with the SSHv2 specification and only for use within the SSHv2 protocol.

- For SRTP, the AES-GCM IV is internally generated at its entirety randomly using an Approved DRBG, whose seed is generated inside the module's physical boundary. Per *NIST SP 800-38D*, the IV length is 96 bits.

## 2.8 EMI / EMC

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

## 2.9 Self-Tests

The module performs power-up self-tests, conditional self-tests, and critical function tests. These tests are described in the sections that follow.

### 2.9.1 Power-Up Self-Tests

The SBC 5400 Session Border Controller performs the following self-tests at power-up to verify the integrity of the firmware images and the correct operation of the FIPS-Approved algorithm implementations:

- Firmware integrity tests using SHA-256 (for OS, SonusDB, EMA, Crypto Library, and SBC firmware components)

- Media Processor algorithm tests:
  - AES encrypt KAT[61] (128-bit, CBC mode)
  - AES decrypt KAT (128-bit, CBC mode)
  - HMAC KAT (with SHA-1)
  - Triple-DES encrypt KAT (3-key, CBC mode)

[60] IV – Initialization Vector
[61] KAT – Known Answer Test

  - Triple-DES decrypt KAT (3-key, CBC mode)

- Crypto Library algorithm tests:
  - AES encrypt KAT (128-bit, ECB mode)
  - AES decrypt KAT (128-bit, ECB mode)
  - AES GCM encrypt KAT (256-bit)
  - AES GCM decrypt KAT (256-bit)
  - Triple-DES encrypt KAT (3-key, ECB mode)
  - Triple-DES decrypt KAT (3-key, ECB mode)
  - HMAC KAT (with SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512)
  - CTR_DRBG KAT (256-bit AES)
  - RSA sign/verify KAT (2048-bit)
  - DSA sign/verify PCT[62] (2048-bit)
  - ECDSA sign/verify PCT (curve K-233)
  - KAS-ECC-SSC Primitive "Z" Computation KAT
  - KAS-FFC-SSC Primitive "Z" Computation KAT
  - SRTP KDF KAT
  - SSH KDF KAT
  - TLS KDF KAT

**NOTE**: The firmware integrity tests using SHA-256 utilize (and thus test) the full functionality of the SHA-256 algorithm; thus, no independent KAT for the SHA-256 implementation is required.

The CO or User can run the module's power-up self-tests at any time by power-cycling the module or issuing a reboot command over the module's Management interfaces. Also, the module can be made to perform power-up self-tests by disconnecting and reconnecting power connectors to the module; and for this service, an operator is not required to assume an authorized role.

### 2.9.2 Conditional Self-Tests

The SBC 5400 Session Border Controller performs the following conditional self-tests:

- Continuous Random Number Generator Test (CRNGT) for the DRBG (Crypto Library)
- CRNGT for the NDRNG entropy source (Crypto Library)
- Firmware Load Test using RSA signature verification (for OS, SonusDB, EMA, and SBC)
- RSA sign/verify PCT (Crypto Library)
- DSA sign/verify PCT (Crypto Library)
- ECDSA sign/verify PCT (Crypto Library)

### 2.9.3 Critical Functions Self-Tests

The SBC 5400 Session Border Controller implements the counter-based DRBG (specified in *NIST SP 800-90Arev1*) as its random number generator. This specification requires that certain critical functions be tested conditionally to ensure the security of the DRBG. The following critical function tests (performed at power-up) are implemented by the cryptographic module:

[62] PCT - Pair-wise Consistency Test

- Instantiate Critical Function Test
- Generate Critical Function Test
- Reseed Critical Function Test
- Uninstantiate Critical Function Test

### 2.9.4 Self-Test Failure Handling

Upon failure of the conditional firmware load test, the module enters a “Soft Error” state and disables all access to cryptographic functions and CSPs. This is a transitory error state, during which the error status is recorded to the system log file and/or event audit log file. Upon failure of this self-test, the CO may choose to reject or continue with the firmware load. Rejecting the load will abort the load process, clear the error condition, and the module will continue normal operations with the currently-loaded firmware. Choosing to continue will load the firmware, clear the error condition, and the module will continue operating with the currently-loaded firmware until the next reboot.

Upon failure any other power-up self-test, conditional self-test, or critical function test, the module will go into a “Critical Error” state and disable all access to cryptographic functions and CSPs. All data outputs are inhibited, and a permanent error status will be recorded to the system log file and/or event audit log file. The task that invoked the failed self-test will be suspended, and the current operation will not complete. The management interfaces will not respond to any commands while the module is in this state. The CO must reboot the module to clear the error condition and return to a normal operational state.

## 2.10 Mitigation of Other Attacks

This section is not applicable. The module does not claim to mitigate any attacks beyond the FIPS 140-2 Level 2 requirements for this validation.

# 3. Secure Operation

The SBC 5400 Session Border Controller meets overall Level 2 requirements for FIPS 140-2. The sections below describe how to ensure that the module is running securely. Please note that physical access to the module shall be limited to authorized operators only.

## 3.1 Initial Setup

The SBC 5400 Session Border Controller is delivered in an uninitialized factory state, and requires CO action to set it up as a FIPS-recognized module and operate in an Approved mode of operation

Physical access to the module shall be limited to the Crypto Officer. The CO shall be responsible for performing all initial setup activities, including configuring the platform and installing the SBC application software. For detailed guidance regarding the use of the module's management interfaces for accomplishing these activities, please see the SBC Core 7.2.x Documentation webpage on Sonus' online Documentation and Support Portal and refer to the following document entries:

- EMA User Guide
- CLI Reference Reference

The following sections provide references to step-by-step instructions for the installation of the SBC 5400 device and software, as well as the steps necessary to configure the module for its FIPS-Approved mode of operation.

### 3.1.1 Hardware Receipt, Installation, and Commissioning

The Crypto Officer shall receive the module from Ribbon via trusted couriers (e.g. United Parcel Service, Federal Express, and Roadway). On receipt, the Crypto Officer must check the package for any irregular tears or openings. If any such damage exists, the CO shall indicate that on the shipping document of the carrier and contact Ribbon Communications, Inc. immediately for instructions. The CO shall also retain the packing list, making sure all the items on the list are present (including all the components of the universal rack mount kit that is shipped with the module).

To setup the SBC 5400, the CO must follow the instructions found under the online document entry "Installing SBC 5400 Hardware", which provides detailed guidance for installing rack mount kits, mounting the SBC chassis, attaching the front bezel, connecting cables and power, and powering on the SBC.

Once these steps have been completed, the SBC hardware is considered to be installed and commissioned.

### 3.1.2 Tamper-Evident Label Application

Before the product can be considered a FIPS-recognized module, the CO must place tamper-evident labels on the module as described in the information provided below. These vendor-branded serialized labels are supplied by Ribbon via a FIPS Physical Security Kit (part no. 550-06508) provided with the appliance upon delivery.

To apply the labels, the appliance surfaces must first be cleaned with isopropyl alcohol in the area where the tamper-evident labels will be placed. Prior to affixing the seals, the front bezel must be attached.

The module ships with six (6) tamper-evident labels. The CO shall place three (3) labels on the appliance as follows:

1. Label 1 shall be placed at the bottom left corner of the module so that it is affixed to the front bezel and the chassis bottom (see Figure 6).
2. Label 2 shall be  placed on the top left of the module so that it is affixed to the front bezel and the top cover (see Figure 7).
3. Label 3 is placed on the back over the module so that it is affixed to the top cover and the rear of the chassis (see Figure 8).

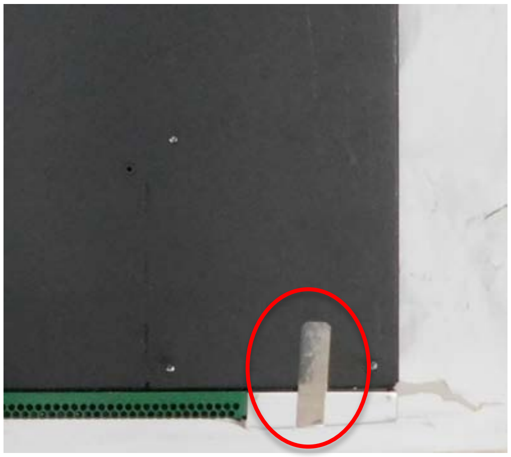

**Figure 6 – Label 1 Placement (Bottom of Chassis)**

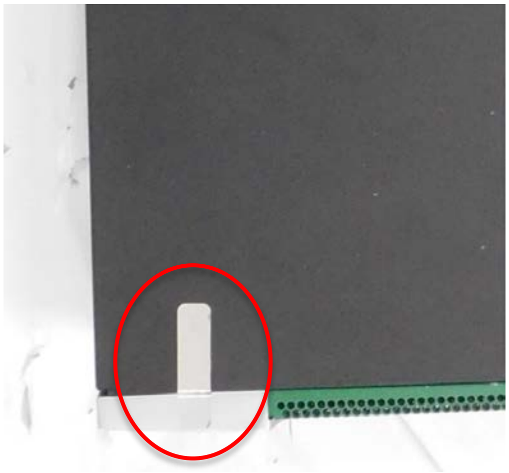

**Figure 7 – Label 2 Placement (Top Cover)**

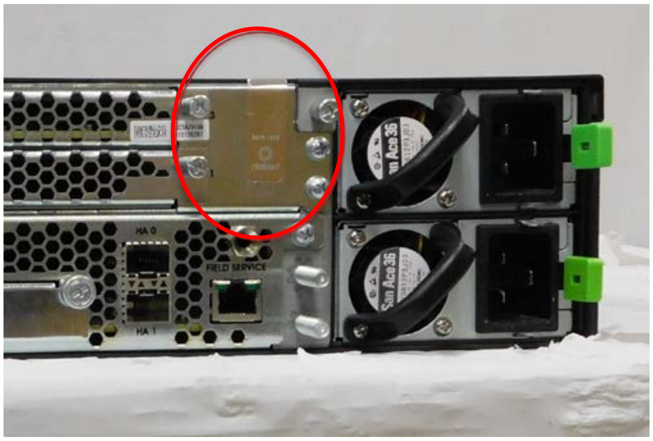

**Figure 8 – Label 3 Placement (Rear of Chassis)**

After applying the labels, allow at least 24 hours for the label adhesive to cure.

Once label application is complete, the CO may then proceed with power-up, installation, and configuration of the module application software.

### 3.1.3 Application Software Installation and Configuration

The next steps are to configure the management interfaces and to install the SBC application software. The CO must follow the instructions under the online document entry "Installing SBC 5400 Software", which provides detailed guidance for general configuration of the platform and installion of the application software.

Once the network settings are correctly configured, the product is considered to be a FIPS-recognized module, and the CO can then perform the procedure in section 3.1.4 in this document to configure the module for operating in Approved mode.

### 3.1.4 Enabling FIPS-Approved Mode

To set the module into its FIPS mode of operation, the CO may use the CLI or the EMA.

To enable FIPS mode 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. Execute the following commands:

```
> configure private
% 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 oam snmp version v3only
% set system admin <system name> fips-140-2 mode enabled
% commit
```

To enable FIPS mode using the EMA, the CO shall complete the following procedure (note that the EMA does not include all of the commands necessary to enable FIPS mode; the CLI must be used to complete the procedure):

1. Log in to the EMA using the default username “`admin`” and password “`admin`”.
2. Using the EMA menu bar, 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”.
5. Click **Save** to save the changes.
6. Using the EMA menu bar, navigate to **All -> Profiles -> Security -> EMA TLS Profile**. The **EMA TLS Profile** window is displayed, with the **EMA TLS Profile List** pane.
7. Select the radio button corresponding to the defaultEmaTlsProfile. The **Edit Selected EMA TLS Profile** pane is displayed.
8. Set the fields V1_0 and V1_1 to “Disabled”. Set the field V1_2 to ”Enabled”.
9. Click **Save** to save the changes.
10. Using the EMA menu bar, navigate to **All -> OAM -> Snmp**. The **Snmp** window is displayed, with the **Edit Snmp** pane.
11. Set the Version field to “V3only”.
12. Click **Save** to save the changes.
13. Log in to the CLI, and execute the following commands:

```
% set system admin <system name> fips-140-2 mode enabled
% commit
```

**NOTE:** To ensure correct functioning and compliance with this Security Policy, module operators must use phones that support TLS v1.2.

Setting the module into FIPS mode will accomplish the following actions:

- All SSH keys will be regenerated.
- Encryption keys used by the system Configuration Database will be regenerated.
- All persistent CSPs stored on the system will be zeroized.

After completion and confirmation of the above steps, the system will reboot. After this reboot, and on all subsequent reboots, the module will be in its FIPS-Approved mode of operation.

## 3.2 Crypto Officer Guidance

The Crypto Officer is responsible for initialization and security-relevant configuration and management of the module. Once installed, commissioned, and configured, the Crypto Officer is responsible for maintaining and monitoring the status of the module to ensure that it is running in its FIPS-Approved mode. Please refer to Section 3.1.4 for guidance that the Crypto Officer must follow for the module to be considered running in a FIPS-Approved mode of operation.

For additional details regarding the management of the module, please refer to the appropriate entries under Ribbon’s SBC Core 7.2.x Documentation webpage.

## 3.2.1 Restoring Service to EMA

After FIPS mode is enabled, the CO must follow the procedures below to install new TLS certificates for EMA (running in Platform Manager mode) to be operational. This ensures that the keys to be used in FIPS mode are established while operating in FIPS mode. Importing of CA and SBC certificates is also addressed in the "Restoring EMA in Platform Mode" section of the online document entry "Enabling SBC for FIPS 140-2 Compliance".

### *3.2.1.1 Import New CA Certificate*

To a import new CA certificate, the CO must perform the following steps:

1. Import a new CA certificate by executing the following commands via the CLI:

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

### *3.2.1.2 Install New SBC Key and Certificate*

The SBC's TLS key and certificate can be generated externally or internally. To install an externally-generated SBC key and certificate, the CO must perform the following steps:

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 by executing the following commands via the CLI:

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

Alternatively, to install an locally-generated SBC key and certificate, the CO must perform the following steps:

1. Generate a certificate signing request (CSR) by executing the following commands via the CLI:

```
> 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 an appropriate CA.
3. Transfer the certificate to the SBC and save it as `/opt/sonus/external/<filename>.pem`.
4. Install certificate by executing the following commands via the CLI:

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

### 3.2.2 Default CO Password Use

The SBC 5400 supports multiple Crypto Officers. This role is assigned when the first CO logs into the system using the default username (“admin”) and password (“admin”). The CO is required to change the default password as part of initial configuration.

Immediately following the initial login on the EMA, module operators are prompted to change the default password. On the CLI, module operators shall use the “change-password” command immediately following the first login.

### 3.2.3 Physical Inspection

For the module to operate in its Approved mode, the tamper-evident labels must be placed by the CO role as specified in section 3.1.2. Per FIPS 140-2 Implementation Guidance 14.4, the CO is also responsible for the following:

- securing and having control at all times of any unused seals
- direct control and observation of any changes to the module where the tamper-evident labels are removed or installed to ensure that the security of the module is maintained during such changes and that the module is returned to its Approved state

The CO is also required to periodically inspect the module for evidence of tampering at intervals specified per end-user policy. The CO must visually inspect the tamper-evident labels for tears, rips, dissolved adhesive, and other signs of attempted tampering. If evidence of tampering is found during periodic inspection, the CO must zeroize the keys and re-image the module before bringing it back into operation.

To request additional labels, the CO can contact Ribbon Customer Service. The CO must be sure to include contact information and the shipping address, as well as the appliance serial number.

### 3.2.4 On-Demand Self-Tests

The power-up self-tests are automatically performed at power-up. The CO may initiate the power-up self-tests by issuing the reboot command or power-cycling the module.

Using the CLI, rebooting the module is accomplished using the following command:

```
% request system admin <systemName> restart
```

Using the EMA, rebooting the module is accomplished by navigating to **All -> System -> Admin -> <systemName> -> Admin Commands -> restart** on the SBC main screen.

Using the EMA in Platform Management Mode, rebooting the module is accomplished by navigating to **Administration -> System Administration -> Platform Management -> Reboot Platform** on the SBC main screen.

### 3.2.5 Zeroization

There are many CSPs within the module’s cryptographic boundary including symmetric keys, private keys, public keys, and login passwords hashes. CSPs reside in multiple storage media including the SDRAM and system SSD.

The default Crypto Officer password is zeroized and replaced when the appliance is re-imaged. All other operator-passwords are zeroized by the CO entering an all-zero password value using the EMA or CLI. All ephemeral keys used by the module are zeroized on reboot, power cycle, or session termination. Keys and CSPs on the SSD of the module can be zeroized by using commands via EMA or CLI. The zeroization of the CDB Key renders other keys and CSPs stored in the non-volatile memory of the CDB useless, effectively zeroizing them. The public key used for the firmware load test is stored in a file in the flash file system, and cannot be zeroized. Reinstallation of the firmware also erases all the volatile and non-volatile keys and CSPs from the module.

Using the CLI, keys and CSPs are zeroized using the following command:

```
% request system admin <systemName> zeroizePersistentKeys
```

Using the EMA, keys and CSPs are zeroized by navigating to **All -> System -> Admin -> <systemName> -> Admin Commands -> zeroizePersistentKeys** on the SBC main screen.

### 3.2.6 Monitoring Status

At any point in time, an authorized operator can access the module via the CLI or the EMA and determine the FIPS mode status of the module. FIPS mode status can be viewed by issuing the following command on the CLI:

```
% show configuration system admin <systemName> fips-140-2 mode
```

When running in Approved mode, the command will return the following message:

```
mode enabled
```

The FIPS mode status of the module can also be viewed using EMA by navigating to **All -> System ->Admin -> Users and Application Management -> Fips 140-2** from the SBC main screen.

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 FIPS-Approved mode. 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 consult the online document entry “SBC Core Troubleshooting Guide” to resolve the issues. If the problems cannot be resolved through these resources, Sonus customer support should be contacted.

### 3.2.7 Firmware Upgrades

To upgrade the module’s application firmware (including the ConnexIP OS and BIOS[63] firmware), the CO shall complete the following procedure:

1. Download the SBC application package from the SalesForce customer portal to the local folder on a PC or remote server.

[63] BIOS – Basic Input/Output System

2. Validate the SBC MD5 checksum using the checksum calculator.
3. Launch the EMA in Platform Management Mode.
4. Upload the desired SBC application package to SBC server using the **Upload Files** tab.
5. Stop the SBC application using **Admin -> Stop Application**. Confirm the stop operation with the CO credentials.
6. Navigate to **SW Install -> Upgrade SBC Application** tab. Select the **SBC Application Version** to upgrade and click **Next**.
7. Confirm the upgrade by providing CO credentials and click **Upgrade.** The upgrade process starts on the SBC and displays the upgrade status on **View Application Upgrade Log** screen.
8. Launch the EMA and log on using the default credentials (change the password if logging for the first time).
9. Verify the SBC application status on the **Administration -> System Administration -> Platform Management** screen.
10. Verify the new OS and SBC application versions in **Monitoring -> Dashboard -> System and Software Info**.

For additional details regarding the module's firmware upgrade process, please refer to the Sonus Support online document entry "Upgrading SBC Application in Standalone Configuration".

## 3.3 User Guidance

The User does not have the ability to configure sensitive information on the module, with the exception of their password. The User must be diligent to pick strong passwords, and must not reveal their password to anyone. Additionally, the User should be careful to protect any secret or private keys in their possession.

## 3.4 Additional Guidance and Usage Policies

This sections notes additional policies below that must be followed by module operators:

- As noted above, operator access to the BMC is provided over two external ports: an RS-232 serial port and a 1 Gbps Ethernet port (called the BMC Field Service Port). The CO must use these ports in order to accomplish the module's initial setup and configuration as described in section 3.1.1 above. Beyond this, the BMC's external ports shall not be used while the module is operational. Use of the BMC's external ports is prohibited while the module is operating in its FIPS-Approved mode.

- Only the CO can create other operators and configure the SBC module to operate in FIPS-Approved mode.

- Password complexities can be configured by the Crypto Officer. All operators shall follow the complex password restrictions. The password may contain any combination of minimum eight letters (upper- and lower-case), numbers, and special characters allowing for a total of 95 possible characters. A password shall have:
    - Between 8 and 24 characters
    - At least one digit
    - At least one lower-case letter
    - At least one upper-case letter
    - At least one special character

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

- 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. The encryption key is established using PBKDF2 as specified in *NIST SP 800-132*. To ensure that the certificate file can be properly decrypted and installed once sent to the module, the encryption algorithm used on the external workstation must be 128-bit AES or 3-key Triple-DES in CBC mode, and the salt length used on the external workstation as input to the PBKDF2 must be 128 bits. Additionally, module operators will need to enter the same passphrase that was used on the external workstation in order to derive the appropriate decrypting key.

- Module operators shall only use RSA keys providing at least 112 bits of encryption strength for signing certificate requests.

- The module allows for the loading of new firmware, and employs an Approved message authentication technique to test its intgrity. However, to maintain an Approved mode of operation, only FIPS-validated firmware can be loaded and executed. Any operation of the module after loading non-validated firmware constitutes a departure from this Security Policy.

  Additionally, the module allows the loading of a new firmware image even if the firmware load test was failed. In this scenario, the module will continue to execute using the existing firmware image; the new image would not be loaded for execution until the next device reboot. Any operation of the module after loading a firmware image that failed the load test is outside the scope of this Security Policy.

- 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 assurances for the keys being used. The following 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*)

- The EMA (running in Platform Manager mode) provides a checkbox that a module operator can use to continue with a firmware upgrade after a failed load test. The Crypto Officer shall ensure that the checkbox remains unchecked while the module is operating in its Approved mode. Any operation of the module after loading unverified firmware constitutes a departure from this Security Policy.

- To ensure that remote authentication is performed over a secured link, the CO shall set the authentication method to PEAP[64]/MS-CHAPv2[65] when configuring the module for RADIUS authentication. For RADIUS authentication configuration guidance via the CLI and the EMA, please refer to the online document entries "Radius Authentication - CLI" and "Radius Authentication – Radius Server", respectively.

- Per *FIPS 140-2 IG* A.13, the CO shall ensure that the module performs no more than 2^16 encryptions with a given Triple-DES key.

- If a RADIUS server is used, an encrypted link using RADIUS over IPsec or TLS should be used.

[64] PEAP – Protected Extensible Authentication Protocol

[65] MS-CHAPv2 – Microsoft Challenge-Handshake Protocol version 2

# 4. Acronyms

Table 13 provides definitions for the acronyms used in this document.

**Table 13 – Acronyms**

| Acronym | Definition |
|---|---|
| AC | Alternating Current |
| ACL | Access Control List |
| AES | Advanced Encryption Standard |
| ANSI | American National Standards Institute |
| ASCII | American Standard Code for Information Interchange |
| BMC | Baseboard Management Controller |
| CA | Certificate Authority |
| CBC | Cipher Block Chaining |
| CCCS | Canadian Centre for Cyber Security |
| CDR | Call Data Record |
| CFB | Cipher Feedback |
| CKG | Cryptographic Key Generation |
| CLI | Command Line Interface |
| CMP | Cryptographic Media Processor |
| CMVP | Cryptographic Module Validation Program |
| CO | Crypto Officer |
| CSP | Critical Security Parameter |
| CSR | Certificate Signing Request |
| CTR | Counter |
| CVL | Component Validation List |
| DC | Direct Current |
| DDOS | Distributed Denial of Service |
| DES | Data Encryption Standard |
| DMZ | Demilitiarized Zone |
| DNS | Domain Name System |
| DOS | Denial of Service |
| DRBG | Deterministic Random Bit Generator |
| DSA | Digital Signature Algorithm |
| DSP | Digital Signal Processor |
| EC | Elliptic Curve |

| Acronym | Definition |
|---|---|
| ECC | Elliptic Curve Cryptography |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| EMA | Embedded Management Application |
| EMC | Electromagnetic Compatibility |
| EMI | Electromagnetic Interference |
| ENUM | E.164 Number Mapping |
| FFC | Finite Field Cryptography |
| FIPS | Federal Information Processing Standard |
| Gbps | Gigabits per second |
| GCM | Galois/Counter Mode |
| GUI | Graphical User Interface |
| HA | High Availability |
| HMAC | (Keyed-) Hash Message Authentication Code |
| HTTPS | Hypertext Transfer Protocol Secure |
| IEEE | Institute of Electrical and Electronics Engineers |
| IKE | Internet Key Exchange |
| IP | Internet Protocol |
| IPMI | Intelligent Platform Management Interface |
| IPsec | Internet Protocol Security |
| IV | Initialization Vector |
| KAS | Key Agreement Scheme |
| KAT | Known Answer Test |
| KDF | Key Derivation Function |
| LED | Light Emitting Diode |
| MAC | Message Authentication Code |
| Mbps | Megabits per second |
| MKEK | Master Key Encrypting Key |
| N/A | Not Applicable |
| NAT | Network Address Translation |
| NDRNG | Non-Deterministic Random Number Generator |
| NIST | National Institute of Standards and Technology |
| NTP | Network Time Protocol |
| OS | Operating System |
| PBKDF2 | Password-Based Key Derivation Function 2 |
| PKCS | Public Key Cryptography Standards |
| QoS | Quality of Service |

| Acronym | Definition |
|---|---|
| RADIUS | Remote Authentication Dial In User Service |
| RAM | Random Access Memory |
| RNG | Random Number Generator |
| RSA | Riverst, Shamir, and Adleman |
| RTCP | Real-time Transport Control Protocol |
| RTP | Real-time Transport Protocol |
| SBC | Session Border Controller |
| SDRAM | Synchronous Dynamic Random Access Memory |
| SFP | Small Form-Factor Pluggable |
| SFTP | SSH (or Secure) File Transfer Protocol |
| SHA | Secure Hash Algorithm |
| SIP | Session Initiation Protocol |
| SNMP | Simple Network Management Protocol |
| SP | Special Publication |
| SRTCP | Secure Real-Time Transport Control Protocol |
| SRTP | Secure Real-Time Transport Protocol |
| SSC | Shared Secret Computation |
| SSD | Solid State Drive |
| SSH | Secure Shell |
| TCP | Transport Control Protocol |
| TLS | Transport Layer Security |
| UDP | User Datagram Packet |
| USB | Universal Serial Bus |
| VOIP | Voice Over Internet Protocol |

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
http://www.corsec.com