Sonus Networks, Inc.

SBC Software Edition Session Border Controller

Software Version: R5.1.2

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.9

Prepared for:

Sonus Networks, Inc.
4 Technology Park Drive

Westford, MA 01886
United States of America

Phone: +1 855 GO SONUS
www.sonus.net

Prepared by:

Corsec Security, Inc.
13921 Park Center Road
Suite 460
Herndon, VA 20171
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 Software Edition Session Border Controller from Sonus Networks, Inc. (Sonus). This Security Policy describes how the SBC Software Edition 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 Canada's Communications Security Establishment (CSE) 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 1 FIPS 140-2 validation of the module. The SBC Software Edition Session Border Controller is referred to in this document as the SBC SWe, 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 Sonus website (www.sonus.net) contains information on the full line of products from Sonus.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) contains contact information for individuals to answer technical or sales-related questions for 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 areas 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.

2. SBC SWe Session Border Controller

2.1 Overview

Sonus Networks, Inc. (hereafter referred to as Sonus) is a leader in IP[1] networking with proven expertise in delivering secure, reliable and scalable next-generation infrastructure and subscriber solutions. The Sonus 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 Sonus SBC Software Edition 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" design 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 Sonus Virtual Network Function (VNF) Manager.

The SBC SWe features the same code base, resiliency, media transcoding, and security technology found in Sonus' hardware-based Sonus 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 Sonus' 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
- 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]

[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
[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

Management of the SBC Software Edition Session Border Controller is accomplished using the following tools:

- Embedded Management Application (EMA), a graphical user interface (GUI) for provisioning, maintaining, and administering the SBC SWe from any Web browser.
- A command line interface (CLI) accessible over a network via an SSH client terminal emulator.

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.

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

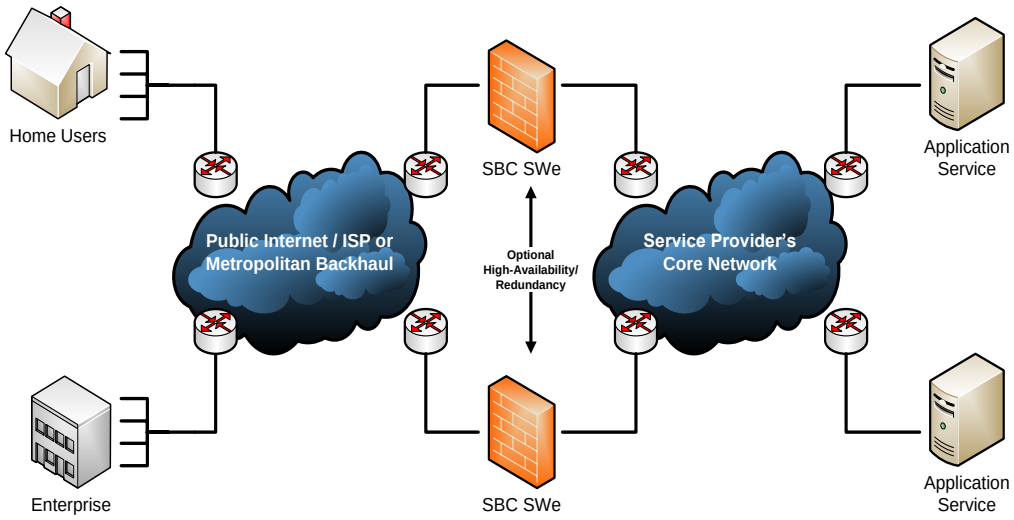

Figure 1 – Typical Deployment of SBC SWe in a Network

The SBC Software Edition 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	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	2
4	Finite State Model	1

[14] SNMPv3 – Simple Network Management Protocol version 3
[15] CDR – Call Detail Records

Section	Section Title	Level
5	Physical Security	N/A[16]
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC[17]	1
9	Self-tests	1
10	Design Assurance	2
11	Mitigation of Other Attacks	N/A

2.2 Module Specification

The SBC Software Edition Session Border Controller is a software cryptographic module with a multiple-chip standalone embodiment that meets overall Level 1 FIPS 140-2 requirements. It executes as a virtual appliance, running on Sonus' proprietary ConnexIP operating system (OS) in a virtual machine (VM) on a general-purpose computer (GPC) hardware platform.

The virtualization layer is provided by VMware ESXi 6.0 (also referred to throughout this document as the hypervisor). The module interacts directly with the hypervisor, which runs directly on the GPC hardware without the need of a host OS. The module was tested and found compliant on an HPE ProLiant DL380 Gen9 server with dual Intel Xeon processors running Sonus's ConnexIP OS over a VMware ESXi 6.0 hypervisor.

The module includes two software cryptographic libraries that provide basic cryptographic functionalities and support secure networking protocols. The software libraries for the module are:

- Sonus Cryptographic Library v3.0 – used for basic cryptographic primitives and for support of SSH and SSL/TLS protocols
- Sonus SRTP Cryptographic Library v1.0 – used for SRTP encryption and key generation

The SBC SWe implements the FIPS-Approved cryptographic algorithms listed in Table 2 below.

Table 2 – Approved Algorithms

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
#4685	AES[18]	FIPS PUB 197	CBC, CTR[19]	128	encryption/decryption
#4686	AES	FIPS PUB 197	CBC, CFB1[20], CFB8, CFB128, ECB	128, 192, 256	encryption/decryption
		NIST SP 800-38D	GCM[21]	128, 256	encryption/decryption

[16] N/A – Not applicable
[17] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility
[18] AES – Advance Encryption Standard
[19] CTR – Counter
[20] CFB – Cipher Feedback
[21] GCM – Galois Counter Mode

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
Vendor Affirmed	CKG[22]	NIST SP 800-133	-	-	symmetric key generation
#1327	CVL[23]	NIST SP 800-56Arev2	ECC CDH Primitive	All NIST-recommended curves	shared secret computation
#1380	CVL	NIST SP 800-135rev1	SNMPv3, SRTP[24], SSH, TLSv1.2	-	key derivation
#1588	DRBG[25]	NIST SP 800-90A	CTR-based	128	deterministic random bit generation
#1202	ECDSA[26]	FIPS 186-4	PKG, PKV	All NIST-recommended curves	key pair generation and verification
			SigGen, SigVer	All NIST-recommended curves	digital signature generation and verification
#3100	HMAC[27]	FIPS 198-1	SHA-1[28]	-	message authentication
#3101	HMAC	FIPS 198-1	SHA-1, SHA-224, SHA-256. SHA-384, SHA-512	-	message authentication
#2556	RSA[29]	FIPS 186-4	SigGenPKCS1.5[30]	2048	digital signature generation
			SigVerPKCS1.5	1024, 2048	digital signature verification
			KeyGen9.31	2048	key pair generation
#3835	SHS[31]	FIPS 180-4	SHA-1	-	message digest
#3836	SHS	FIPS 180-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	-	message digest
#2490	Triple-DES[32]	NIST SP 800-67	TCBC, TCTR	Keying option 1	encryption/decryption
#2491	Triple-DES	NIST SP 800-67	TECB, TCBC	Keying option 1	encryption/decryption

Note: *No parts of the SNMP, SRTP, SSH, and TLS protocols, other than the KDF, have been tested by the CAVP.*

The module uses the FIPS-Approved counter-based DRBG specified in NIST SP 800-90A to generate cryptographic keys. The resulting symmetric key or a 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.

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

[22] CKG – Cryptographic Key Generation
[23] CVL – Component Validation List
[24] SRTP – Secure Real-Time Transport Protocol
[25] DBRG – Deterministic Random Bit Generator
[26] ECDSA – Elliptic Curve Digital Signature Algorithm
[27] HMAC – (keyed-) Hashed Message Authentication Code
[28] SHA – Secure Hash Algorithm
[29] RSA – Rivest Shamir Adleman
[30] PKCS – Public Key Cryptography Standard
[31] SHS – Secure Hash Standard
[32] DES – Data Encryption Standard

Table 3 – Allowed Algorithms

Algorithm	Caveat	Use
Diffie-Hellman	key establishment methodology provides 112 bits of encryption strength	key agreement
Elliptic Curve Diffie-Hellman (ECDH)	key establishment methodology provides between 112 and 256 bits of encryption strength	key agreement
MD5[33]	-	TLSv1.2 protocol handshake
NDRNG[34]	-	seeding for the FIPS-Approved DRBG
RSA	key establishment methodology provides 112 bits of encryption strength	key transport

2.2.1 Physical Cryptographic Boundary

As a virtual appliance, the software module has no physical characteristics; however, the module 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 is responsible for mapping the module's virtual interfaces to the host server's physical interfaces.

The physical boundary of the cryptographic module is defined by the hard enclosure around the host server on which it runs. For this validation, the module will be tested on an HP ProLiant DL380 Gen9 server, which consists of a motherboard, two Intel Xeon processors, random access memory (RAM), a hardware case, a power supply, and interface ports. Other devices may be attached to the hardware platform such as a monitor, keyboard, mouse, DVD[35] drive, fixed disk drive, printer, video adapter, audio adapter, or network adapter.

Figure 2 shows diagrams the hardware components of the server used for testing (the dashed line surrounding the hardware components represents the module's physical cryptographic boundary, which is the outer case of the server), and identifies the hardware with which the processors interface.

[33] MD5 – Message Digest 5
[34] NDRNG – Non Deterministic Random Number Generator
[35] DVD – Digital Video Disc

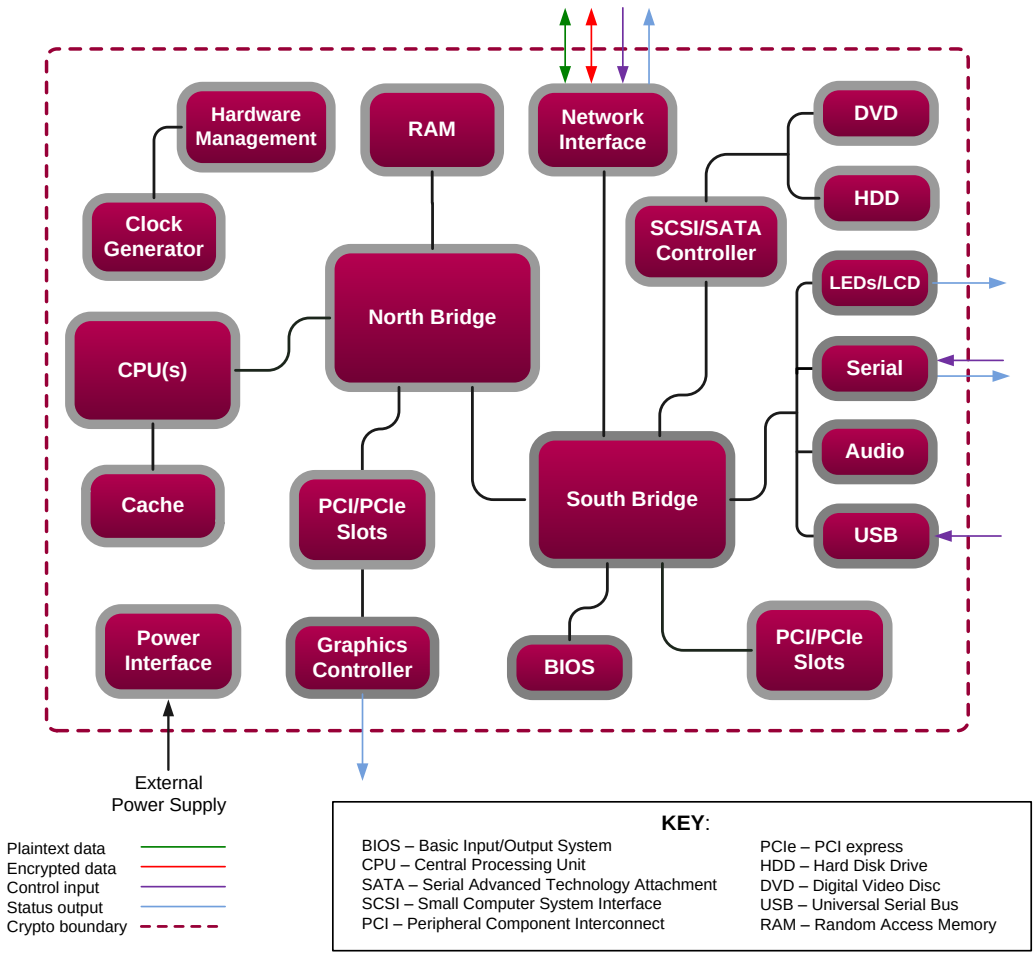

Figure 2 – Block Diagram of the Host Server

The module's physical cryptographic boundary is further illustrated by the black dotted line in Figure 3 below.

2.2.2 Logical Cryptographic Boundary

The logical cryptographic boundary of the module (shown by the red dotted line in Figure 3 below) consists of the Sonus SBC SWe application and Sonus' ConnexIP operating system acting as the guest OS.

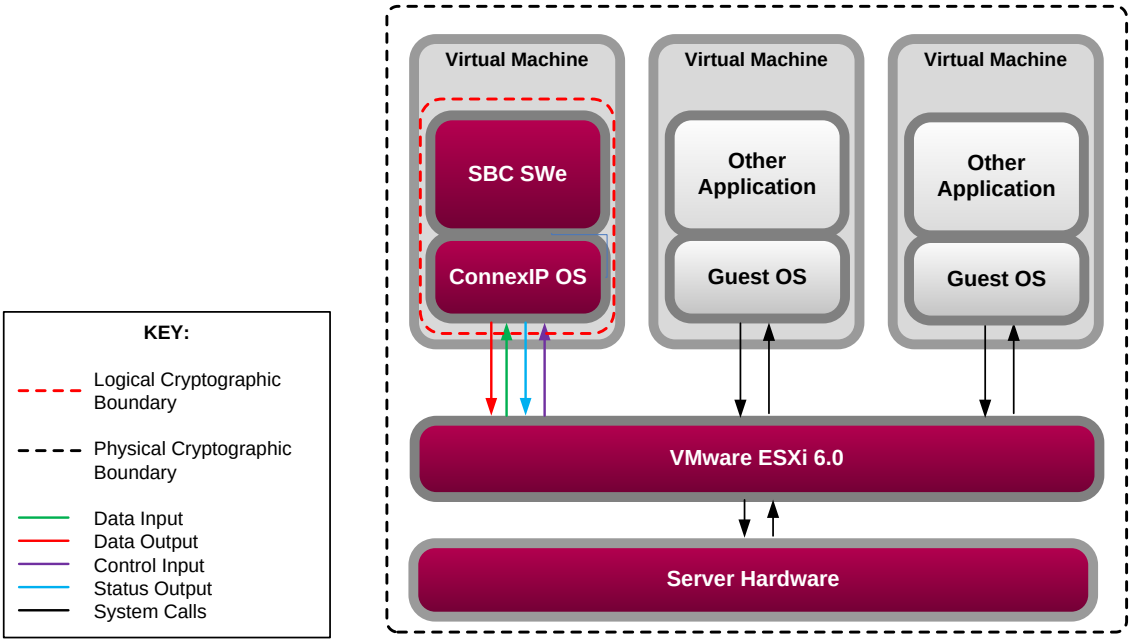

Figure 3 – Sonus SBC SWe Cryptographic Boundaries

2.3 Module Interfaces

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

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

As a virtual appliance, the SBC SWe 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 is intended to run on a GPC. It was tested and validated on an HPE DL380 Gen9 server. The following is a list of physical interfaces implemented on the tested host server:

- Ethernet ports
- Keyboard port
- Mouse port
- Serial port
- Video port
- LEDs[36]
- AC[37] Power port

[36] LED – Light-Emitting Diode
[37] AC – Alternating Current

The mapping of the module's logical interfaces in the software to FIPS 140-2 logical interfaces is described in Table 4 below.

Table 4 – SBC SWe Interface Mappings

Physical Port/Interface	Virtual Port/Interface	FIPS 140-2 Logical Interface
Host server Ethernet (10/100/1000) ports	Virtual Ethernet ports	• Data Input • Data Output • Control Input • Status Output
Host server keyboard port	Virtual keyboard port	• Control Input
Host server mouse port	Virtual mouse port	• Control Input
Host server serial port	Virtual serial port	• Data Input • Control Input
Host server video connector	Virtual video connector	• Status Output
Host server LEDs	Virtual LEDs	• Status Output
Host server power interface	N/A	• Power

2.4 Roles and Services

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, and is the only operator that can install and configure the module for FIPS-Approved mode of operation. The CO 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. Users can also change their own passwords.

2.4.2 Operator Services

Descriptions of the services available to the Crypto Officer role and User role are provided in the Table 5 below. The keys and CSPs listed in Table 5 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 5 – 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[38] rules, NTP[39] and DNS[40] servers	Command and parameters	Command response/ Status output	None
Configure Crypto Suite Profile	✓		Select crypto suites for SRTP, SRTCP, and SIP communication	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	Password – R/W/X
Manage user sessions	✓		Terminate User sessions	Command and parameters	Command response/ Status output	TLS Session Key – R/X SSH Session Key – R/X

[38] ACL – Access Control List
[39] NTP – Network Time Protocol
[40] DNS – Domain Name System

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Change password	✓	✓	Modify existing login passwords	Command and parameters	Command response/ Status output	Password – R/W
Load certificate	✓		Load new certificates	Command	Command response/ Status output	Certificate Load Key – R CA[41] Public Key – R/W TLS Private Key – R/W TLS Public Key – R/W TLS Peer Public Key – R/W 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
View Event Log	✓		View event status messages	Command	Command response/ Status output	None
Zeroize keys	✓		Zeroize all keys and CSPs	Command	Command response/ Status output	All ephemeral and persistent keys and CSPs – W
Upgrade firmware	✓		Load new software and performs an integrity test using an RSA digital signature	Command	Command response/ Status output	Software Load Authentication Key – R/X
Perform keying of CDB[42] 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

[41] CA – Certificate Authority
[42] CDB – Configuration Database

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Establish TLS session	✓	✓	Establish a web session using TLS protocol	Command	Command response/ Status output	Diffie-Hellman Public Key – R/X Diffie-Hellman Private Key – X ECDH Public Component – R/X ECDH Private Component – X TLS Private Key – W/X TLS Public Key – W/X TLS Peer Public Key – R/X TLS Pre-Master Secret – W/X TLS Master Secret – W/X TLS Session Key – R/W/X TLS Authentication Key – W/X
Establish SSH session	✓	✓	Establish a remote session using SSH protocol	Command	Command response/ Status output	Diffie-Hellman Public Key – R/X Diffie-Hellman Private Key – X ECDH Public Component – R/X ECDH Private Component – X SSH Private Key – W/X SSH Public Key – W/X SSH Peer Public Key – R/X SSH Shared Secret – W/X SSH Session Key – R/W/X SSH Authentication Key – W
Establish SRTP session	✓	✓	Establish a SIP/TLS session using SRTP protocol	Command	Command response/ Status output	SRTP Master Key – R/X SRTP Session Key – W/X SRTP Authentication Key – W/X
SNMPv3 traps		✓	Provides system condition information	None	Status output	SNMPv3 Session Key – R/W/X 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 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 6 lists the services for which the operator is not required to assume an authorized role. None of these services disclose or substitute cryptographic keys and CSPs or otherwise affect the security of the module.

Table 6 – Additional Services

Service	Description	Input	Output	CSP and Type of Access
Zeroize	Zeroize keys and CSPs	Remove power from the host server	Status output	All ephemeral keys and CSPs – W
Perform On-Demand Self-Tests	Perform power-up self-tests on demand	Cycle power on the host server	Status output	All ephemeral keys and CSPs – W
Authenticate	Used for operator logins to the module	Command	Status output	Password – X RADIUS Shared Secret – W/X 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.

Each operator must authenticate using the correct username/password. The module also supports RSA digital certificate authentication of operators during Web GUI/HTTPS (TLS) access. Table 7 lists the authentication mechanisms used by the module. The strength calculations provide minimum strength based on password policy described in Section 3.4.

Table 7 – 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 1: (95^8), or 1: 6,634,204,312,890,625. The fastest network connection over Ethernet Interface supported by the module is 1000 Mbps. Hence, at most (10 ×10^8 × 60 = 6 × 10^{10} =) 60,000,000,000 bits of data can be transmitted in one minute. Therefore, the probability that a random attempt will succeed or a false acceptance will occur in one minute is 1 : [95^8 possible passwords / ((6 ×10^{10} bits per minute) / 64 bits per password)] 1: (95^8 possible passwords / 937,500,000 passwords per minute) 1: 7,076,484; 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 1:2^{112} or 1: 5.19 x 10^{33}. The fastest network connection supported by the module over Ethernet interfaces is 1000 Mbps. Hence at most (100 ×10^7 × 60 = 6 × 10^{10} =) 60,000,000,000 bits of data can be transmitted in one minute. Therefore, the probability that a random attempt will succeed or a false acceptance will occur in one minute is 1: (2^{112} possible keys / ((6 × 10^{10} bits per minute) / 112 bits per key)) 1: (2^{112} possible keys / 535,714,285 keys per minute) 1: 96.92 × 10^{23}; which is less than 100,000 as required by FIPS 140-2.

The feedback of authentication data to a user 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 providing 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 cryptographic module is a software module and does not include physical security mechanisms. Therefore, as per Section G.3 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*, requirements for physical security are not applicable.

2.6 Operational Environment

The operational environment of the module does not provide a general-purpose operating system (OS) to the user.

The SBC SWe runs on Sonus' proprietary ConnexIP OS, which acts as the guest OS on top of the virtualization layer. The virtualization layer is provided by VMware's ESXi hypervisor v6.0. The VMware hypervisor runs directly on the server's hardware, with no need for an underlying operating system. Only the module's signed image can be executed, and all software upgrades are digitally-signed.

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

2.7 Cryptographic Key Management

The module supports the CSPs described in Table 8 below.

Table 8 – Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
Config Database (CDB) Key	Triple-DES 192-bit key	Generated internally via FIPS-Approved DRBG	Never exits the module	Plaintext in SSD	When re-keyed over CLI or EMA; When appliance is re-imaged; Upon command via CLI or EMA	Encryption of RSA and ECDSA private keys and preshared secrets for RADIUS in CDB
CA Public Key	2048-bit RSA key	Generated externally and imported in DER[43] file format	Never exits the module	Plaintext in RAM		Verification of Certificate Authority signatures
ECDH Private Component	Private component of ECDH protocol	Generated internally	Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	Establishment of TLS/SSH session keys
ECDH Public Component	Public component of ECDH protocol	[for the module] Generated internally [for a peer] Entered into the module (in certificate form) in plaintext	[for the module] Exits the module in plaintext form [for a peer] Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	Establishment of TLS/SSH session keys
Diffie-Hellman Private Key	2048-bit DH key	Generated internally	Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	Generation of SSH/TLS shared secrets

[43] DER – Distinguished Encoding Rules

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
Diffie-Hellman Public Key	2048-bit DH key	[for the module] Generated internally [for a peer] Entered into the module (in certificate form) in plaintext	[for or the module] Exits the module in plaintext form [for a peer] Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	Generation of SSH/TLS shared secrets
SSH Private Key	2048-bit RSA key	Generated internally via FIPS-Approved DRBG	Never exits the module	Encrypted in the CDB on SSD	Command via CLI or EMA	Authentication during SSH session negotiation
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	Command via CLI or EMA	Authentication during SSH session negotiation **1024-bit key is used for signature verification only**
SSH Peer Public Key	1024/2048-bit key	Imported in plaintext	Never exits the module	Plaintext in the CDB on SSD	Command via CLI or EMA	Authentication during session negotiation
SSH Shared Secret	Shared secret	Derived internally via DH/ECDH shared secret computation	Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	Derivation of the SSH Session Key and SSH Authentication Key
SSH Session Key	128/256-bit AES or 192-Triple-DES bit key	Derived internally via SSH KDF	Never exits the module	Plaintext in RAM	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	Plaintext in RAM	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 public key [for authentication using ECDSA certificates] All NIST-recommended ECDSA curves	[for local-internal certificates] Generated internally via FIPS-Approved DRBG	Never exits the module	Encrypted in the CDB on SSD	Command via CLI or EMA	Authentication during TLS key negotiation
TLS Public Key	[for authentication using RSA certificates] 2048-bit RSA public key [for authentication using ECDSA certificates] All NIST-recommended ECDSA curves	[for local-internal certificates] Generated internally via FIPS-Approved DRBG	Exits the module via digital certificate in plaintext form	Plaintext in the CDB on SSD	Command via CLI or EMA	Authentication during TLS key negotiation
TLS Peer Public Key	[for authentication using RSA certificates] 2048-bit RSA public key [for authentication using ECDSA certificates] All NIST-recommended curves	Imported in certificate form in plaintext	Never exits the module	Plaintext in the CDB on SSD	Command via CLI or EMA	Certificate-based authentication during TLS key negotiation

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
TLS Pre-Master Secret	[for RSA cipher suites] 384-bit random value [for DH/ECDH cipher suites] DH/ECDH shared secret	[for RSA cipher suites and module acting as client] Generated internally via FIPS-Approved DRBG [for RSA cipher suites and module acting as server] Generated externally, imported in encrypted form via RSA key transport [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 [for RSA cipher suites and module acting as server] Never exits the module [for DH/ECDH cipher suites] Never exits the module	Plaintext in RAM	Upon module reboot; 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	Plaintext in RAM	Upon module reboot; Upon session termination	Derivation of the TLS Session Key and TLS Authentication Key
TLS Session Key	128/256-bit AES key	Derived internally using the TLS Master Secret via TLS KDF	Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	Encryption and decryption of TLS session packets
TLS Authentication Key	160-bit (minimum) HMAC key	Derived internally using the TLS Master Secret via the TLS KDF	Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	Authentication of TLS session packets
SRTP Master Key	128-bit shared secret	Generated externally, imported in encrypted form via a secure SIP/TLS session	Exits in encrypted form	Plaintext in RAM	Upon module reboot; Upon session termination	Peer Authentication, Session and Authentication keys derivation for SRTP session

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
SRTP Session Key	128-bit AES-CTR key	Generated internally using Master Key	Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	Encryption or decryption during SRTP session
SRTP Authentication Key	160-bit HMAC key	Generated internally using Master Key	Never exits the module	Plaintext in RAM	Upon module reboot; 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
DRBG Seed	256-bit value	Generated internally using entropy input string	Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	Generation of random number
Entropy Input String	512-bit value	Continually polled from various host server resources to accrue entropy by the NDRNG	Never exits the module	Plaintext in RAM	Upon module reboot; Upon session termination	
SFTP Private Key	2048-bit RSA key	Generated internally via FIPS-Approved DRBG	Never exit the module	Stored in the CDB on SSD – encrypted for the certificates; stored outside CDB on SSD – plaintext for SSH	Upon command via CLI or EMA	Used for SFTP key negotiation
SFTP Public Key	1024/2048-bit RSA key	The module's 2048-bit public key is generated internally; public key of a peer enters the module in plaintext	The module's 2048-bit public key exits the module in plaintext form; public key of a peer never exits the module	Plaintext in the CDB on SSD	Upon command via CLI or EMA	Used for SFTP key negotiation **1024-bit key is used for signature verification only**
SNMPv3 Session Key	AES-CFB 128-bit or Triple-DES 192-bit	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	Encrypting SNMPv3 packets

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
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
Crypto Officer password	Alphanumeric string (minimum of eight characters)	Initial password generated internally using FIPS-Approved DRBG, password changes entered into module via a console port or over SSH	Initially generated password provided to the CO on CLI/EMA over encrypted session, changed password never exits the module	Plaintext[44] on SSD and in RAM	When the password is updated with a new one	Authenticating the Crypto Officer to the module
User password	Alphanumeric string (minimum of eight characters)	Initial password generated internally using DRBG, password changes entered into module via a console port or over SSH	Initially generated password provided to the User on CLI/EMA over encrypted session, changed password never exits the module	Plaintext on SSD and in RAM	When the password is updated with a new one	Authenticating the User to the module
Software Load Authentication Key	2048-bit key RSA public key	Embedded in release image	Never exits the module	Stored in flash memory or on spinning media (depending upon server configuration)	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 software load package

[44] CO and User passwords are obfuscated by the operating system and stored on the SSD. They are temporarily loaded into the memory in obfuscated form for comparison during a login.

The module uses a FIPS-Approved SP 800-90A CTR_ DRBG to generate cryptographic keys. The module actively requests seeding material for the DRBG from /dev/random, which exists within the module's logical cryptographic boundary. The entropy source is a blocking NDRNG, ensuring that the module's request for seed material is only fulfilled when sufficient entropy (minimum 512 bits) can be provided.

2.8 EMI / EMC

The module's host server 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 Software Edition Session Border Controller performs the following self-tests at power-up to verify the integrity of the software images and the correct operation of the FIPS-Approved algorithm implementations:

- Software integrity tests using HMAC SHA 256
- IPP algorithm tests:
  - AES encrypt KAT[45]
  - AES decrypt KAT
  - Triple-DES encrypt KAT
  - Triple-DES decrypt KAT
  - HMAC SHA-1 KAT
- Crypto Library algorithm tests:
  - AES encrypt KAT
  - AES decrypt KAT
  - AES GCM encrypt KAT
  - AES GCM decrypt KAT
  - Triple-DES encrypt KAT
  - Triple-DES decrypt KAT
  - HMAC SHA-1, HMAC SHA-224, HMAC SHA-256, HMAC SHA-384, HMAC SHA-512 KAT
  - SP 800-90A CTR_DRBG KAT
  - RSA signature generation KAT
  - RSA signature verification KAT
  - ECDSA pair-wise consistency test
  - ECDH primitive "Z" computation test

Note: HMAC KATs with SHA-1 and SHA-2 utilize (and thus test) the full functionality of the SHA-1 and SHA-2 algorithms; thus, no independent KATs for SHA-1 and SHA-2 implementations are required.

[45] KAT – Known Answer Test

The CO or User can run the module's power-up self-tests at any time by issuing a reset/reboot command over its management interfaces. Also, the module can be made to perform power-up self-tests by rebooting or power-cycling the module's VM (for this service, the operator is not required to assume an authorized role).

2.9.2 Conditional Self-Tests

The SBC Software Edition Session Border Controller performs the following conditional self-tests for the Crypto Library:

- Continuous Random Number Generator Test (CRNGT) for the DRBG
- CRNGT for the NDRNG entropy source
- Software load test using verification of an RSA signature and HMAC SHA-256 digest (for OS, SonusDB, EMA, and SBC)
- RSA sign/verify KAT
- ECDSA pair-wise consistency test
- ECDH public key assurance test

2.9.3 Critical Functions Self-Tests

The SBC Software Edition Session Border Controller implements the SP 800-90A CTR_DRBG as its random number generator. The SP 800-90A specification requires that certain critical functions be tested conditionally to ensure the security of the DRBG. Therefore, the following critical function tests are implemented by the cryptographic module:

- SP 800-90A Instantiate Critical Function Test
- SP 800-90A Generate Critical Function Test
- SP 800-90A Reseed Critical Function Test
- SP 800-90A Uninstantiate Critical Function Test

2.9.4 Self-Test Failure Handling

Upon failure of the conditional software 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 software load. Rejecting the load will abort the load process, clear the error condition, and the module continues normal operations with the currently-loaded software. Choosing to continue will load the software, clear the error condition, and the module will continue operating with the currently-loaded software 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 restart the module (i.e. reboot the host server) 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 1 requirements for this validation.

3. Secure Operation

The sections below describe how to place and keep the SBC Software Edition Session Border Controller in its Approved mode of operation.

3.1 Installation and Setup

The module is available as a software package that includes both the application software and the operating system. It can be downloaded from the Sonus Salesforce Customer Portal (login credentials are required for portal access).

The Crypto Officer is responsible for all initial setup activities, including configuring virtual machine, installing the guest operating sytem, and installing the SBC SWe application software. For detailed guidance regarding these activities, please see the SBC Core 5.1.x Documentation webpage on Sonus' online Documentation and Support Portal and refer to the following document entries:

- Installing SBC SWe on Virtual Platforms
- EMA User Guide
- CLI Command Reference

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

3.1.1 Software Installation

To setup the SBC SWe, the CO must follow the instructions found under the document entry "Installing SBC SWe on Virtual Platforms", which provides detailed guidance for downloading, verifying, installing, and running the SBC SWe software in a VMware virtual environment.

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

3.1.2 Application Configuration

The next steps are to configure the module's application software. The CO must follow the instructions under the document entry "Installing SBC SWe on Virtual Platforms", which provides detailed guidance for changing system settings and configuring various module features.

Once the module is installed with network settings properly configured, the Crypto Officer must then enable FIPS mode.

3.1.3 FIPS-Approved Mode Configuration and Status

To enable FIPS mode, the CO shall execute the following commands via CLI after logging in using the default username "admin" and password "admin" (note that enabling FIPS mode using the EMA is not currently supported):

```
> 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
```

After completion of these steps, the module's VM will reboot. After this reboot, and on all subsequent reboots, the module will be operating 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.

3.2.1 Management

Once installed 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 3.1.3 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 Sonus' SBC Core 5.1.x Documentation webpage.

3.2.2 Default CO Password Use

The module provides multiple sets of default credentials for performing various portions of the CO role's responsibilities.

- After installating the ConnexIP OS on the virtual machine, the VM instance will reboot and a ConnexIP login screen will be displayed. The CO must first login to the OS using the default username "linuxadmin" and password "sonus".
- Then the CO must login as root user with the default password "sonus1" in order to perform the initial network configurations.
- First-time access to the module's CLI requires the CO to login using the the default username "admin" and password "admin".

In all cases, the CO shall change all default passwords immediately after their initial use.

3.2.3 Loading TLS Certificates

To support TLS, the module implements three types of certificates:

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

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

- 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.

After enabling FIPS mode, the Crypto Officer shall install new TLS certificates via the EMA. The CO shall ensure that only 2048-bit RSA keys are used for signing CSRs for local-internal certificates.

3.2.4 Zeroization

There are many CSPs within the module's cryptographic boundary including symmetric key, private keys, public keys, and login passwords hashes. CSPs reside in multiple storage media including the SDRAM and system SSD. All ephemeral keys are zeroized on module reboot, power cycle, or session termination. Keys and CSPs that are stored 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 software load test is stored in a file in the flash file system, and cannot be zeroized. Reinstallation of the software 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 using the following steps:

On SBC main screen, navigate to **All -> System ->Admin -> <systemName> -> Admin Commands -> zeroizePersistentKeys**

3.2.5 Monitoring Status

On the first power up, the module is, by default, in an unconfigured operational state. During initial configuration and setup, the module is explicitly set to operate in the FIPS-Approved mode of operation. An authorized operator can access the module via the CLI or the EMA and determine the FIPS-Approved mode of the module.

At any point of time, the status of the module (i.e. FIPS mode status) can be viewed by issuing the following command on the CLI:

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

The 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 operational, 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 contact Sonus Customer Support.

3.3 User Guidance

The User role does not have the ability to configure sensitive information on the module, with the exception of their password. The User must be diligent to select strong passwords, and must not reveal their password to anyone. Additionally, User role operators 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:

- Only the CO can create other operators.
- 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 AES GCM encryption/decryption shall be established.
- The module allows for the loading of new software, and employs an Approved message authentication technique to test the new software's integrity. However, to maintain an Approved mode of operation, the CO shall ensure that only FIPS-validated software is loaded. Any operation of the module after loading non-validated software is outside the scope of this Security Policy.
- The platform manager provides a checkbox that a module operator can use to continue with a software 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 software is outside the scope of this Security Policy.
- The CO shall ensure that the module performs no more than 2^{28} encryptions with a given Triple-DES key.

3.5 Non-Approved Mode of Operation

When in the operational state, the module can alternate on a service-by-service basis 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 completion of the non-Approved service.

3.5.1 Security Functions

The module includes the following non-Approved algorithm(s) that shall only be used in a non-Approved mode of operation:

- IKE v1/v2 KDF (non-compliant)

3.5.2 Roles

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

3.5.3 Services

Table 9 below lists the service(s) available in the non-Approved mode of operation.

Table 9 – Non-Approved Services

Service	Operator		Description
	CO	User	
Establish IPsec Session (non-compliant)	✓	✓	Establish remote session using IPsec protocol

4. Acronyms

Table 10 provides definitions for the acronyms used in this document.

Table 10 – 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
CA	Certificate Authority
CBC	Cipher Block Chaining
CDR	Call Data Record
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CLI	Command Line Interface
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CSE	Communications Security Establishment
CSP	Critical Security Parameter
CSR	Certificate Signing Request
CTR	Counter
CVL	Component Validation List
DC	Direct Current
DDOS	Distributed Denial of Service
DER	Distinguished Encoding Rules
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

Acronym	Definition
EC	Elliptical Curve
ECC	Elliptical Curve Cryptography
ECDSA	Elliptical Curve Digital Signature Algorithm
EMA	Embedded Management Application
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
ENUM	E.164 NUmber Mapping
FIPS	Federal Information Processing Standard
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
IKEv1	Internet Key Exchange version1
IP	Internet Protocol
IPsec	Internet Protocol Security
KAT	Known Answer Test
KDF	Key Derivation Function
LED	Light Emitting Diode
MAC	Message Authentication Code
Mbps	Mega-bits per second
MD5	Message Digest 5
MKEK	Master Key Encrypting Key
N/A	Not Applicable
NDRNG	Non-Deterministic Random Number Generator
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OS	Operating System
PKCS	Public-Key Cryptography Standard
RADIUS	Remote Authentication Dial In User Service
RAM	Random Access Memory
RNG	Random Number Generator
RSA	Riverst, Shamir, and Adleman

Acronym	Definition
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
SSD	Solid State Drive
SSH	Secure Shell
TCP	Transport Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
USB	Universal Serial Bus
VM	Virtual Machine
VNF	Virtual Network Function

Prepared by:

Corsec Security, Inc.

13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

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