LifeSize Communications, Inc.

Cryptographic Security Kernel

Software Version: 1.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 1.0

Prepared for:

LifeSize Communications, Inc.
901 S. Mopac Building 3 Suite 300
Austin, Texas 78746
U.S.A.

Phone: +1 (512) 347-9300
Fax: +1 (512) 347-9301
http://www.lifesize.com

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, Virginia 22030
U.S.A.

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

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Cryptographic Security Kernel from LifeSize Communications, Inc. This Security Policy describes how the Cryptographic Security Kernel meets the security requirements of FIPS 140-2 and how to run the module in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 – *Security Requirements for Cryptographic Modules*) 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 Cryptographic Module Validation Program (CMVP) website, which is maintained by National Institute of Standards and Technology (NIST) and Communication Security Establishment Canada (CSEC): http://csrc.nist.gov/groups/STM/index.html.

The Cryptographic Security Kernel is referred to in this document as the cryptographic module, 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 LifeSize website (http://www.lifesize.com) contains information on the full line of products from LifeSize.
- The CMVP Vendor List (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401vend.htm) contains contact information for answers to technical or sales-related questions for the module.

1.3 Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Machine
- Submission Summary
- Other supporting documentation and additional references

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

2 Cryptographic Security Kernel

2.1 Overview

LifeSize Communications, Inc. is a high definition video and audio communication company that provides High Definition video endpoints, infrastructure and management solutions for all types of organizations. The LifeSize lines of video communications products provide affordable, full-motion video and audio telepresence solutions for any size room. LifeSize sells a robust set of product lines. The LifeSize "Room" line consists of the Room, Room 200, and Room 220, and is designed to be deployed in medium-sized conference rooms. The LifeSize "Team" line consists of the Team MP, Team 200, and the Team 220, and is designed to be deployed in small conference rooms or used by small groups of people. The LifeSize "Express" line consists of the Express, Express 200, and the Express 220, and is designed to be deployed at an individual's desk for single-person use.

The LifeSize products are actually systems consisting of several system components (see Figure 1 below for the components of the Room 200 product). At a minimum, the systems will include the following components:

- LifeSize coder/decoder device (codec)
- LifeSize high-definition pan-tilt-zoom camera
- LifeSize phone or "MicPod"
- wireless remote
- power supply
- cables

Figure 1 – LifeSize Room 200 (w/ codec, remote, phone, and camera)

The LifeSize codecs perform the bulk of the audio and video processing, provide the primary administrative and user interfaces, and perform encryption and security functions. All of the LifeSize codecs have a similar architecture (differing in licensed features and available physical ports), and all run the same software image.

Each LifeSize codec has multiple physical interfaces that allow it to be connected to audio/video sources such as cameras, microphones, LifeSize phones, and Ethernet networks. The codecs also implement several management interfaces: an on-screen User Interface (UI) that is displayed on an attached monitor

and manipulated via an infrared remote control; a command-line interface (CLI) via SSH[1] and a serial port; a web-based graphical user interface (GUI) via HTTP[2] and HTTPS[3], and an SNMP[4] interface.

Audio and video data is transported via the Session Initiation Protocol (SIP) or H.323 (an audio/video communications protocol for packet networks), and this data can be encrypted if both sides of the communication agree to use encryption. The codecs, utilizing security functions provided by LifeSize's Cryptographic Security Kernel, use the Advanced Encryption Standard (AES) algorithm to encrypt and decrypt data.

All of the products' cryptographic functionality is provided by the Cryptographic Security Kernel, which is composed of functionality contained in a single object file. The cryptographic module runs on a Linux 2.4 Operating System (OS), executed by a PowerPC processor, and does not modify or become part of the OS kernel.

The Cryptographic Security Kernel is validated at the following FIPS 140-2 Section levels:

Table 1 – FIPS 140-2 Security Levels

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC[5]	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

2.2 Cryptographic boundary

The following sections will define the physical and logical boundary of the Cryptographic Security Kernel.

2.2.1 Physical Cryptographic Boundary

As a software cryptographic module there are no physical security mechanisms implemented per se, the module must rely on the physical characteristics of the host system. The physical cryptographic boundary of the Cryptographic Security Kernel is defined by the hard enclosure around the host system on which it runs. All physical ports are realized by the host system's physical ports and may vary per platform. Figure 1 below depicts the typical LifeSize hardware platform with accompanying physical ports, the dotted blue

[1] SSH – Secure Shell
[2] HTTP – Hypertext Transfer Protocol
[3] HTTPS – Secure Hypertext Transfer Protocol
[4] SNMP – Simple Network Management Protocol
[5] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

line surrounding the module components represents the module's physical cryptographic boundary, while the ports and interfaces exist at the boundary and interface with the various controllers.

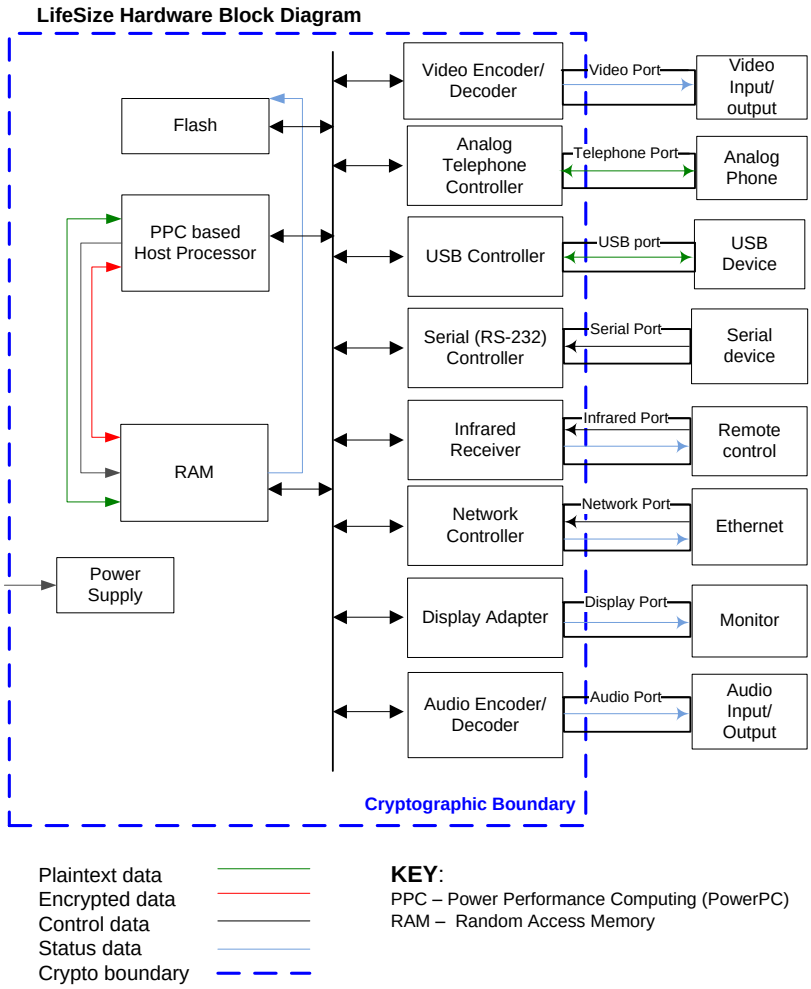

Figure 2 – LifeSize Host System Block Diagram

2.2.2 Logical Cryptographic Boundary

Figure 3 below shows a logical block diagram of the module. The module is a software only cryptographic module running on a Linux operating system. The module's logical cryptographic boundary encompasses all functionality contained within a single object file. This object file is linked at build-time to a shared object, which can be called by host applications to provide cryptographic services.

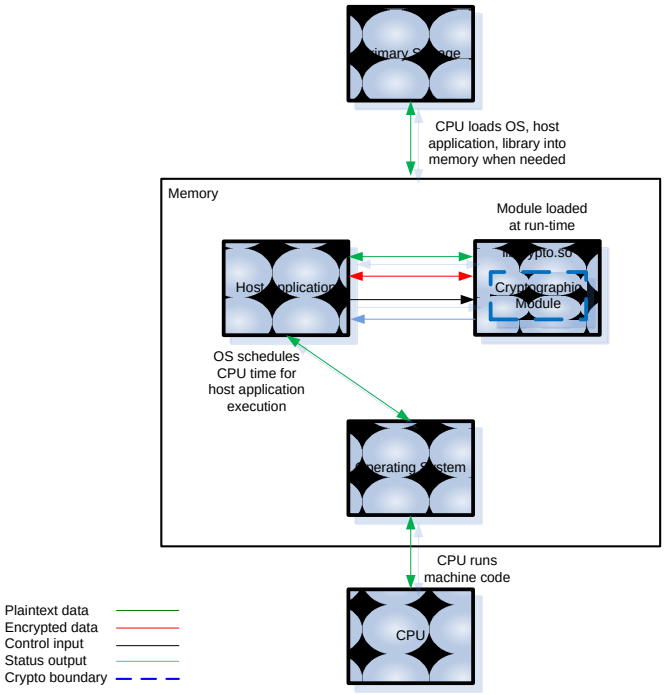

Figure 3 – Logical Block Diagram and Cryptographic Boundary

2.3 Module Interfaces

The module's logical interfaces exist at a low level in the software as an Application Programming Interface (API). The API interface is mapped to the following four logical interfaces:

- Data input
- Data output
- Control input
- Status output

The module features the physical ports of the host system, as depicted in Figure 2. The following is a list of physical interfaces implemented on a host system:

- Video port
- Telephone port
- USB[6] port
- Serial port
- Infrared port (Remote control)
- Network port
- Display port
- Audio port
- AC Power port

As a software module, the module has no physical characteristics. Thus, the module's manual controls, physical indicators, and physical and electrical characteristics are those of the host system.

The FIPS-defined interfaces map to their physical and logical counterparts as described in Table 2 below.

[6] USB – Universal Serial Bus

Table 2 – FIPS Interface Mappings

FIPS 140-2 Interface	Physical Interface	Module Interface (API)
Data Input	• Video port • Telephone port • USB port • Serial port • Infrared port (Remote control) • Network port • Audio port	Function calls that accept, as their arguments, data to be used or processed by the module
Data Output	• Video port • Telephone port • USB port • Serial port • Network port • Display port • Audio port	Arguments for a function that specify where the result of the function is stored
Control Input	• Serial port • Infrared port (Remote control) • Network port	Function calls utilized to initiate the module and the function calls used to control the operation of the module.
Status Output	• Video port • Serial port • Network port • Display port • Audio port	Return values for function calls
Power Input	• AC power port	N/A

2.4 Roles and Services

While the module itself provides no mechanism for the authentication of operators, it supports the following authorized roles: the Crypto-Officer (CO) role and the User role.

Note: The following definitions are used in the "CSP and Type of Access" column in Table 3 and Table 4.

Read - The item is **read** or **referenced** by the service.
Write - The item is **written** or **updated** by the service.
Execute - The item is **executed** by the service. (The item is used as part of a cryptographic function.)

2.4.1 Crypto-Officer Role

The Crypto-Officer role is used for initializing the module and for accessing the module's symmetric/asymmetric encryption/decryption, signature generation/verification, hashing, cryptographic key generation, random number generation, and message authentication functions.

Descriptions of the services available to the Crypto-Officer role are provided in Table 3 below.

Table 3 – Mapping of Crypto Officer Role's Services to Type of Access

Service	Description	CSP and Type of Access
Installation	Installation of the module	None

Service	Description	CSP and Type of Access
Initialization	Perform module initialization	None
Hashing (SHA[7])	Perform hashing operation	None
Message Authentication (HMAC[8])	Perform message authentication services	HMAC SHA-1 key – Execute
Symmetric Encryption/Decryption	Perform encryption/decryption operation	AES[9] symmetric key – Execute TDES[10] symmetric key – Execute
Digital Signature	Perform sign and verify operation	DSA[11] private key – Execute RSA[12] private key – Execute
Key Establishment	Key establishment-supporting mechanism (Diffie-Hellman, RSA)	RSA private key – Execute Diffie-Hellman private key – Execute
Symmetric Key Generation	Generate symmetric keys	AES Symmetric key – Execute TDES Symmetric key – Execute
Asymmetric Key Generation	Generate asymmetric keys	DSA private key – Execute RSA private key – Execute
Pseudo-Random Number Generation	Generate random numbers	Seed key – Execute Seed – Execute
Show Status	Show status of the module	None
Perform Self-Tests	Perform self-tests on demand	None
Zeroization	Zeroize all CSPs	AES Symmetric key – Write TDES Symmetric key – Write DSA private key – Write RSA private key – Write Diffie-Hellman private key – Write HMAC SHA-1 key – Write Seed key – Write

2.4.2 User Role

Like the CO role, the User roles is used to access symmetric/asymmetric encryption/decryption, signature generation/verification, hashing, cryptographic key generation, random number generation, and message authentication functions.

Descriptions of the services available to the User role are provided in Table 4.

Table 4 – Mapping of User Role's Services to Type of Access

Service	Description	CSP and Type of Access
Initialization	Perform module initialization	None

[7] SHA – Secure Hash Algorithm
[8] HMAC – (Keyed) Hash Message Authentication Code
[9] AES – Advanced Encryption Standard
[10] TDES – Triple Data Encryption Standard
[11] DSA – Digital Signature Algorithm
[12] RSA – Rivest Shamir Adleman

Service	Description	CSP and Type of Access
Hashing (SHA)	Perform hashing operation	None
Message Authentication (HMAC)	Perform message authentication services	HMAC SHA-1 key – Execute
Symmetric Encryption/Decryption	Perform encryption/decryption operation	AES Symmetric key – Execute TDES Symmetric key – Execute
Digital Signature	Perform sign and verify operation	DSA private key – Execute RSA private key – Execute
Key Establishment	Key establishment-supporting mechanism (Diffie-Hellman, RSA)	RSA private key – Execute Diffie-Hellman private key – Execute
Symmetric Key Generation	Generate symmetric keys	AES Symmetric key – Execute TDES Symmetric key – Execute
Asymmetric Key Generation	Generate asymmetric keys	DSA private key – Execute RSA private key – Execute
Pseudo-Random Number Generation	Generate random numbers	Seed key – Execute Seed – Execute
Show Status	Show status of the module	None
Perform Self-Tests	Perform self-tests on demand	None
Zeroization	Zeroize all CSPs	AES Symmetric key – Write TDES Symmetric key – Write DSA private key – Write RSA private key – Write Diffie-Hellman private key – Write HMAC SHA-1 key – Write Seed key – Write

2.5 Physical Security

The Cryptographic Security Kernel is purely a software module. As such, it depends on the physical characteristics of the host system and its protection mechanisms. In this case the host system is the hardware provided by LifeSize. Thus, physical security requirements do not apply.

2.6 Operational Environment

The module was tested for FIPS 140-2 validation on Linux 2.4 operating system. For FIPS 140-2 compliance, this is considered to be single user operating system. As such, all keys, intermediate values, and other CSPs remain only in the process space of the operator using the module. The operating system uses its native memory management mechanisms to ensure that outside processes cannot access the process space used by the module.

2.7 Cryptographic Key Management

The module implements the FIPS-Approved algorithms listed in Table 5 below.

Table 5 – FIPS-Approved Algorithm Implementations

Approved Security Function	Certificate Number
Symmetric Key Algorithm	
AES - 128-,192-, 256-bit in ECB[13], CBC[14], CFB[15]8, CFB128 and OFB[16] modes	1123
Triple-DES - 112-, 168-bit in ECB, CBC, CFB8 and OFB modes	820
Secure Hashing Algorithm (SHA)	
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1046
Message Authentication Code (MAC) Function	
HMAC using SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	634
Pseudo Random Number Generator (PRNG)	
ANSI[17] X9.31 Appendix A.2.4 PRNG	626
Asymmetric Key Algorithm	
RSA (X9.31, PKCS #1.5, PSS) sign/verify: 1024-, 1536-, 2048-, 3072-, 4096-bit	532
DSA sign/verify: 1024- bit	365

Additionally, the module utilizes the following non-FIPS-approved algorithm implementations (these algorithms are allowed for use in a FIPS-Approved mode of operation):

- Diffie-Hellman support for key agreement: 1024-bits for key establishment (Caveat: provides 80 bits of encryption strength)

  NOTE: The module does not provide full Diffie-Hellman key agreement, only the Diffie-Hellman algorithm/functionality and primitives.

- RSA key wrapping: 1024-, 2048-, 3072-, 7680-, 15360-bit (key wrapping; key establishment methodology provides between 80 and 256 bits of encryption strength)

[13] ECB – Electronic Codebook
[14] CBC – Cipher-Block Chaining
[15] CFB – Cipher Feedback
[16] OFB – Output Feedback

[17] ANSI – American National Standards Institute

The module supports the following critical security parameters:

Table 6 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
Symmetric keys	AES - 128-, 192-, 256-bit Triple DES - 112-,168-bit	Generated Internally using the FIPS Approved PRNG	None	Resides in plaintext on volatile memory	By API call	Encrypt/decrypt
Asymmetric public key	RSA 1024-,1536-, 2048-, 3072- and 4096-bit public key	Generated Internally using the FIPS Approved PRNG	None	Resides in plaintext on volatile memory	By API call	Key establishment, key wrapping
Asymmetric private key	RSA 1024-,1536-, 2048-, 3072- and 4096-bit private key	Generated Internally using the FIPS Approved PRNG	None	Resides in plaintext on volatile memory	By API call	Key establishment, key wrapping
DSA public key	DSA 1024-bit	Generated Internally using the FIPS Approved PRNG	None	Resides in plaintext on volatile memory	By API call	Sign
DSA private key	DSA 1024-bit	Generated Internally using the FIPS Approved PRNG	None	Resides in plaintext on volatile memory	By API call	Verify
HMAC-SHA-1 key	HMAC SHA-1	None	None	Resides in plaintext on volatile memory	By API call	Generate checksum, software integrity test

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
ANSI X9.31 PRNG seed	16 bytes	None	None	Resides in plaintext on volatile memory	By power cycle	Generate random number
ANSI X9.31 PRNG seed key	16-, 24- or 32 bytes AES key	None	None	Resides in plaintext on volatile memory	By API call	Generate random number

2.7.1 Key Generation

The module uses an ANSI X9.31 Appendix A.2.4 PRNG implementation to generate cryptographic keys. This PRNG is FIPS-Approved as shown in Annex C to FIPS PUB 140-2.

2.7.2 Key Entry and Output

The cryptographic module itself does not support key entry or key output from its physical boundary. However, keys are passed to the module as parameters from applications resident on the host platform via the exposed APIs. The host application using the module is responsible for ensuring that the input or output of secret and private keys is accomplished in encrypted form.

2.7.3 CSP Storage and Zeroization

The module does not persistently store any CSPs. All of the keys and CSPs in Table 6 above reside only on the volatile memory in plaintext and can be zeroized using the destruction method included in the API. This method is capable of overwriting the key in memory with random bytes.

2.8 EMI/EMC

The module is a software module, and depends on the host systems for its physical characteristics. However, the host systems have been tested for, and meet, applicable Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B, Class A of FCC 47 Code of Federal Regulations Part 15. All systems sold in the United States must meet the applicable FCC requirements.

2.9 Self-Tests

The module performs the following self-tests at power-up:

- Software integrity test using HMAC SHA-1
- Cryptographic Algorithm tests
    - AES Known Answer Test (KAT)
    - Triple-DES KAT
    - HMAC KATs
        - HMAC SHA-1
        - HMAC SHA-224
        - HMAC SHA-256
        - HMAC SHA-384
        - HMAC SHA-512
    - SHA KATs
        - SHA-1 (performed as part of the HMAC-SHA-1 known answer test)
        - SHA-224 (performed as part of the HMAC-SHA-224 known answer test)
        - SHA-256 (performed as part of the HMAC-SHA-256 known answer test)
        - SHA-384 (performed as part of the HMAC-SHA-384 known answer test)
        - SHA-512 (performed as part of the HMAC-SHA-512 known answer test)
    - RSA encrypt/decrypt KAT
    - RSA signature generation/verification KAT
    - DSA pairwise consistency test for sign/verify
    - ANSI X9.31 PRNG Appendix A.2.4 KAT

The module performs the following conditional self-tests:

- DSA pairwise consistency test for sign/verify
- Continuous Random Number Generator test (CRNGT)
- RSA pairwise consistency test for sign/verify and encrypt/decrypt

The module enters an error state when a power-up self-test fails. The power-up self-tests run through to completion without interruption, and provide no mechanism for data output. Upon power-up self-test failure, the module will enter a critical error state and will fail to launch in FIPS mode of operation. No data output or cryptographic operations are possible when the module enters the critical error state.

No data output or cryptographic operations are possible when the module enters the soft error state. The module will exit upon encountering all self-test errors, power-on or conditional. The hosting application is responsible for re-invocation of the module. If the error cannot be cleared, the module must be re-installed.

2.10 Design Assurance

LifeSize uses Subversion (SVN) version control system v1.5.4 as the configuration management system. It provides code version control, code sharing and build management. It supports check-in, check-out and merging. It is "wrapped" in a number of scripts to support hourly, daily and production builds. Branches in Subversion are used to isolate project and release development. It can also be locked during different phases to control changes.

The software development cycle is built around the best industry practices which involve release planning, designing, development and software/hardware quality assurance. Every release (major, maintenance, hotfix, and patch) is "tagged" so that it can always be retrieved. Production builds are archived in document control and backups. Release Notification is sent to the Release team including document control, marketing and support.

Additionally, Microsoft Visual SourceSafe (VSS) version 6.0 is used to provide configuration management for the Cryptographic Security Kernel's FIPS documentation. This software provides access control, versioning, and logging.

2.11 Mitigation of Other Attacks

The module does not claim to mitigate any additional attacks in an approved FIPS mode of operation.

3 Secure Operation

The Cryptographic Security Kernel meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in its FIPS-approved mode of operation.

3.1 Initial Setup

t is the Crypto-Officer's responsibility to configure the module according to FIPS-Approved mode. As stated in Section 2.4, an operator can access the application linked to the module through an infrared remote interface. The infrared remote provides the operator with a menu interface.

The sections below describe how to initialize and manage the module in FIPS-Approved mode of operation and how to make secure calls.

3.2 Crypto-Officer Guidance

The following two sections contain the necessary guidance to securely install and administer the cryptographic module.

3.2.1 Initialization

The Crypto-Officer is responsible for initializing and installing the module. The software module will be provided to the Crypto-Officer by LifeSize Communications, Inc.. The module is installed during the process of installing the host application. With the delivered software, the Crypto-Officer also receives detailed documentation on installing, uninstalling, configuring, managing and upgrading the host application.

FIPS Mode

A single initialization call is required to initialize the module in the FIPS-Approved mode of operation. (NOTE: the module does not support a non-FIPS mode). The FIPS mode initialization is performed when the host application invokes function call which returns a "1" for success and "0" for failure. Interpretation of this return code is the responsibility of the host application. The LifeSize application is capable of reporting the error condition to screen when the module fails.

Installation & Initialization

1. Perform the "Initial Configuration" to setup the LifeSize hardware and application as per the LifeSize Installation Guide. These steps are necessary to prepare the hardware platform for interoperating with the module later.

2. Navigate to "Administrator Preferences / Security / General" in the application GUI;

3. Check the box that says "FIPS 140-2" to say enabled, and select "back" a page.

   The hardware device containing the module will then reboot.

4. Navigate to "Administrator Preferences" and verify that the page says "FIPS 140-2 Security Enabled"

5. Ensure that there is no persistently red FIPS icon is in the status below.

The module has now been successfully installed and configured for FIPS 140-2.

(Note: If the red FIPS icon is present on the application status bar, the module has failed a critical self-test and is not operating in the Approved mode.) If rebooting the hardware device does not clear the error, the operator can attempt to reinstall the module.

3.2.2 Management

The Crypto-Officer is responsible for ensuring the module is initialized correctly as per Section 3.2.1 of this document. The module will enter the error state and terminate if any self-test does not pass.

The Crypto-Officer is able to monitor and configure the module via the infrared remote interface. Detailed instructions to manage and troubleshoot the host application are provided in the Administrator's Guide. The Crypto-Officer should monitor the module status regularly for FIPS mode of operation. The Crypto-Officer can check the status of the module by looking at the status bar. If a red FIPS icon is present, then the module failed to initialize. The absence of this icon in conjunction with the "FIPS 140-2 Security Enabled" message in "Administrator Preferences" indicates that the module was successful entering the Approved mode.

3.3 User Guidance

The cryptographic functionality of the module (i.e. the collection of User role services) is listed in Table 4 above.

4 Acronyms

This section describes the acronyms used throughout the document.

Table 7 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
CBC	Cipher-Block Chaining
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CRNGT	Continuous Random Number Generator Test
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ECB	Electronic Codebook
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
HMAC	(Keyed-) Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
ISDN	Integrated Services Digital Network
KAT	Known Answer Test
LED	Light Emitting Diode
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OFB	Output Feedback
OS	Operating System
PC	Personal Computer
PKCS	Public Key Cryptography Standard
PPC	Power Performance Computing (PowerPC)

Acronym	Definition
PRNG	Pseudo Random Number Generator
PSS	Probabilistic Signature Scheme
RAM	Random Access Memory
RSA	Rivest Shamir and Adleman
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SVN	Subversion
USB	Universal Serial Bus
VSS	Visual SourceSafe

Prepared by:
Corsec Security, Inc.

10340 Democracy Lane, Suite 201
Fairfax, Virginia 22030
U.S.A.

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