Recently acquired by Aruba,
a Hewlett Packard Enterprise company

Silver Peak Systems, Inc.

Silver Peak ECOS Cryptographic Library

Software Version: Crypto Library 2020 Version 1.0

FIPS 140-2 Level 1 Non-Proprietary Security Policy

Document Version Number: 1.7

June 8, 2023

Table of Contents

Tables

Figures

Document Revision History

Version	Date	Author	Description
1.0	October 30, 2021	Silver Peak Systems, Inc.	Initial release with ECOS version 8.1.9
1.1	November 8, 2021	Silver Peak Systems, Inc.	Updates to include ECOS version 9.1
1.2	January 26, 2022	Silver Peak Systems, Inc.	Updates for CMVP review comments
1.3	April 13, 2022	Silver Peak Systems, Inc.	Updates for CMVP review comments
1.4	August 10, 2022	Silver Peak Systems, Inc.	Updates for CMVP review comments
1.5	February 16, 2023	Silver Peak Systems, Inc.	Updates for CMVP review comments
1.6	May 15, 2023	Silver Peak Systems, Inc.	Updates for CMVP review comments

Aruba, a Hewlett Packard Enterprise company, acquired Silver Peak Systems, Inc. in 2020.
For more details see HPE Completes Acquisition of SD-WAN Leader Silver Peak.

1 Introduction

This document is the non-proprietary security policy for the Silver Peak ECOS Cryptographic Library, hereafter referred to as the Module. The Module is a software library providing a C language application program interface (API) for use by other processes that require cryptographic functionality. The module is loaded into an existing environment at boot up, and there is no need for specific configuration.

The Module is classified by FIPS 140-2 as a software module, multichip standalone module embodiment. The physical cryptographic boundary is the general-purpose computer on which the module is installed. The logical cryptographic boundary of the Module is the fipscanister object module, a single object module file named fipscanister.o. This is highlighted in the green-dashed box shown in Figure 1. The Module performs no communications other than with the calling application (the process that invokes the Module services) via the API.

The appliances used during testing met Federal Communications Commission (FCC) FCC Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined by 47 Code of Federal Regulations, Part15, Subpart B, Class B, which vacuously meets requirements for Class A in FIPS 140-2 Area 8 (Security Level 1).

Table 1 Intended Level of Security

Area	FIPS Security Area	Security 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	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

Security Levels shown in Table 1 above refer to FIPS PUB 140-2, Table 1
The Module's software version for this validation is **Crypto Library 2020 Version 1.0**.

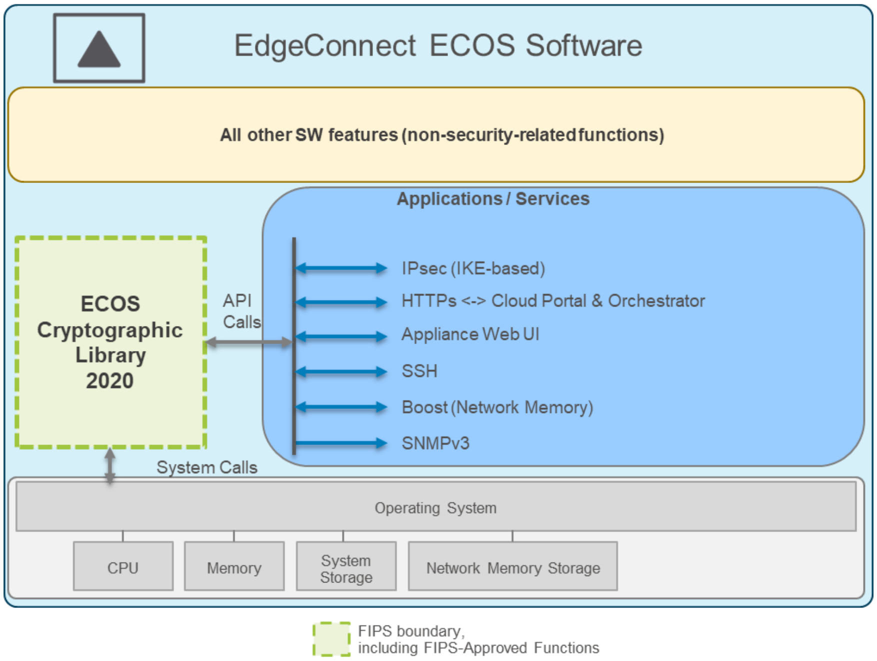
EdgeConnect ECOS Software
All other SW features (non-security-related functions)
Applications / Services
ECOS Cryptographic Library 2020
API Calls
IPsec (IKE-based)
HTTPs <-> Cloud Portal & Orchestrator
Appliance Web UI
SSH
Boost (Network Memory)
SNMPv3
System Calls
Operating System
CPU
Memory
System Storage
Network Memory Storage
FIPS boundary, including FIPS-Approved Functions

Figure 1 ECOS Cryptographic Module Block Diagram

The module is Crypto Library 2020 Version 1.0 which is a module inside the EdgeConnect (ECOS) Software Package.

2 Tested Configurations

FIPS 140-2 conformance testing was performed at Security Level 1. The following configurations were tested by the lab.

Table 2 Tested Configurations

Silver Peak ECOS Cryptographic Library	Tested Appliance	EdgeConnect ECOS Versions	Operational System	Processor
Crypto Library 2020 Version 1.0.	EC-XS	8.1.9	Fedora Core 6 (2.6.38 Kernel)	Intel Atom C3558 (Denverton) With AES-NI enabled and with AES-NI disabled
Crypto Library 2020 Version 1.0.	EC-XS	9.1	Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel Atom C3558 (Denverton) With AES-NI enabled and with AES-NI disabled

The Cryptographic Module meets FIPS 140-2 Level 1 requirements.
Refer to section 7, Operational Environment, for a list of vendor-affirmed operational environments.

3 Ports and Interfaces

The logical interface is a C-language application program interface (API).

Table 3 FIPS 140-2 Logical Interfaces

Logical Interface Type	Description
Control input	API entry point and corresponding stack parameters
Data input	API entry point data input stack parameters
Status output	API entry point return values and status parameters
Data output	API entry point data output stack parameters

As a software module, control of the physical ports is outside module scope. However, when the module is performing self-tests, or is in an error state, all output on the logical data output interface is inhibited. In error scenarios the module returns only an error value (no data output is returned).

The cryptographic module will be used to provide cryptographic functions to client and server applications. When a crypto module is implemented in a server environment, the server application is the user of the cryptographic module. The server application is the single user of the cryptographic module, even when the server application is serving multiple clients, and therefore the module is considered to operate in single-operator mode.

4 Modes of Operation

The Module supports two modes of operation:

- In FIPS-Approved Mode of operation (the FIPS-Approved mode of operation), only approved or allowed security functions with sufficient security strength can be used.
- In non-FIPS mode (the non-Approved mode of operation), non-approved security functions can also be used.

4.1 FIPS Approved Mode of Operation

Invoking FIPS-Approved Mode of operation:

- After initializing the module, the crypto-officer or user will execute the FIPS_module_mode_set(1) command to invoke the FIPS-Approved Mode of operation.
- The module will be in FIPS-Approved Mode when all power-on self-tests have completed successfully and only Approved algorithms in Tables 4 and 5 are invoked.

The following approved cryptographic algorithms are used in FIPS-Approved Mode of operation.

4.1.1 FIPS Approved Cryptographic Functions

Table 4 FIPS Approved Cryptographic Functions

CAVP Cert # With AES-NI Enabled	CAVP Cert # With AES-NI Disabled	Algorithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
C2122 A1957	C2209 A1957	AES	SP 800-38A FIPS 197, SP 800-38F	CBC, CFB1, CFB128, CFB8, CTR (EXT Only), ECB, OFB	128, 192, 256	Data Encryption/ Decryption
A1958	N/A	AES	SP 800-38A FIPS 197	CBC	128, 192, 256	Data Encryption/ Decryption
C2163 A1957	C2214 A1957	AES-GCM	SP800-38D	GCM Note 1	128, 192, 256	Data Encryption/ Decryption
C2123 A1957	C2210 A1957	CVL Partial DH	SP 800-56Ar3	ECC	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521	ECC CDH Component Primitive
C2136 A1957	C2212 A1957	DRBG	SP 800-90A	Counter Hash based HMAC based		Deterministic Random Bit Generation Note 2
C2126 A1957	C2211 A1957	HMAC	FIPS 198-1	HMAC-SHA-1 HMAC-SHA-224 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	HMAC-SHA1 (112-bit min key) HMAC-SHA224 (128-bit min key) HMAC-SHA256 (128-bit min key) HMAC-SHA384 (192-bit min key) HMAC-SHA512 (256-bit min key)	Message Authentication
C2121 A1957	C2208 A1957	Secure Hash Standard (SHS)	FIPS 180-4	Secure Hash Algorithm (SHA) SHA-1 SHA-224 SHA-256 SHA-384		Message Digest

CAVP Cert # With AES-NI Enabled	CAVP Cert # With AES-NI Disabled	Algorithm	Standard	Mode/ Method	Key Lengths, Curves or Moduli	Use
				SHA-512		
C2161 A1957	C2215 A1957	RSA	FIPS 186-4	PKCS1 v1.5 SHA-1 Note 3 SHA-224 SHA-256 SHA-384 SHA-512	RSA KeyGen (186-4) 2048, 3072 RSA SigGen (186-4) 2048, 3072 RSA SigGen (186-2) 4096 RSA SigVer (186-2) 1024, 1536 Note 3, 2048, 3072, 4096	Digital Signature Generation and Verification Key Generation
C2135 A1957	C2213 A1957	CVL TLS 1.2 IKEv1 SSH SNMP	SP 800-135			Key Derivation Note 4
CKG (vendor affirmed)			SP 800-133r2 Cryptographic Key Generation			Key Generation Note 5

Note 1: The module's AES-GCM implementation complies with IG A.5 scenario 2. 96-bit IVs are generated per SP 800-38D section 8.2.2. See paragraph below regarding AES GCM compliance with I.G. A.5.

Note 2: The module contains an approved DRBG that receives a LOAD command. No assurance of the minimum strength of generated keys. 256 bits is the minimum number of bits of entropy believed to have been loaded.

Note 3: SHA-1 and modulus size 1536 are only used in RSA Signature Verification for legacy use only (186-2).

Note 4: No parts of these protocols, other than the Key Derivation Function (KDF), have been tested by the CAVP and CMVP. KDF algorithms previously were listed under Component Validation List (CVL); this is kept for backwards compatibility.

Note 5: The module directly uses the unmodified output of the DRBG for key generation.

AES GCM compliance with I.G. A.5

The module does not generate the AES GCM (Galois Counter Mode) key, and therefore, the calling application is responsible for providing the AES GCM key.

In approved mode, users of the module must not utilize GCM with an externally generated IV (Initialization Vector). Operator shall meet this requirement by not passing an IV as an argument to the GCM API for service "Symmetric digest". The module will automatically generate the IV using the DRBG.

The module's implementation of AES-GCM is used together with an application that executes outside of the module's cryptographic boundary. The 96-bit GCM IV is generated internally, the module enforces the use of an approved DRGB in line with Section 8.2.2 of SP 800-38D. Per IG A.5, in the event module power is lost and restored the consuming application must ensure that any of its AES-GCM keys used for encryption or decryption are re-distributed.

4.1.2 Non-FIPS Approved but Allowed Cryptographic Functions.

"Non-FIPS Approved but Allowed cryptographic functions" are allowed to be used in a FIPS approved mode of operation. They are categorized as "non-FIPS approved" because they are not explicitly included *Annex A: Approved Security Functions for FIPS PUB 140-2, Security Requirements for Cryptographic Modules, June 10, 2019.*

The Algorithms listed in Table 5 below are "allowed" because they are covered in Implementation Guidance (IG) as follows:

- RSA Key Wrapping

Table 5 Non-FIPS Approved but Allowed Cryptographic Functions

Algorithm	Caveat	Use
RSA Key Wrapping using 2048 / 3072 bits key	Provides 112 or 128 bits of encryption strength.	Used for key establishment such as in TLS handshake (the library does not directly support the TLS protocol).

4.2 Non-Approved Mode of Operation

4.2.1 Cryptographic Functions/Algorithms that are NOT Allowed in FIPS-mode of operation.

Calling algorithms with Modes, Key Lengths, Curves, and/or Moduli with values outside of those specified in Tables 4 and 5 will result in the module no longer being in FIPS mode.

For example, calling RSA with SigGen Probabilistic Signature Scheme (PSS) will result in the module no longer being in FIPS mode.

See Table 6 for all Non-FIPS Approved Algorithms.

EC DH keys are not to be generated by the module and shall only be input to the module by the calling application. Once the module is out of FIPS mode, in order to get it back to FIPS mode, the module must be unloaded from memory via a reboot. This will result in the zeroization of all keys and CSPs in memory.

Table 6 Non-FIPS Approved Algorithms

Function	Algorithm	Mode/ Method/Options
Encryption, Decryption and CMAC	TDES CMAC-TDES	3-Key TDES TECB, TCBC, TCFB, TOFB; CMAC-TDES generate and verify
Encryption, Decryption and CMAC	AES CMAC-AES	XTS; CCM; GCM; CMAC-AES generate and verify
Digital Signature and Asymmetric Key Generation	RSA	SigGenPSS, SigVerPSS (2048/3072/4096 with all SHA-2 sizes)
Digital Signature and Asymmetric Key Generation	DSA	All DSA modes are not allowed
Digital Signature and Asymmetric Key Generation	ECDSA	All ECDSA modes are not allowed

5 Critical Security Parameters and Public Keys

The table below describes cryptographic keys and Critical Security Parameters (CSPs) used by the module.

Table 7 Cryptographic Keys and CSPs

Description/Usage	Type	Generation/Establishment	Input/Output	Key to Entity	Storage	Zeroization
Keys used for **AES** encryption/decryption. **Key** sizes are 128, 192, and 256 bits.	Symmetric encryption/decryption keys AES CBC CFB1 CFB128 CFB8 OFB ECB CTR	SP 800-90A CTR_DRBG; As per SP 800-133r2 Section 4. Generated by the DRBG, is performed as per the “Direct Generation” of Symmetric Keys which is an Approved key generation method. Stored in RAM, until the zeroization function is called or the application is unloaded.	Input: in plaintext by the calling application Output: N/A	calling application user or CO	Plaintext in RAM	Zeroized by calling the zeroization service or by unloading the module.
AES GCM Key Key sizes are 128, 192, and 256 bits	AES GCM	SP 800-90A CTR_DRBG; As per SP 800-133r2 Section 4. Key generation is performed as per the “Direct Generation” of Symmetric Keys which is an Approved key generation method	Input: in plaintext by the calling application Output: N/A	calling application user or CO	Plaintext in RAM	Zeroized by calling the zeroization service or by unloading the module.
DRBG Seed Seed length = key length + 128 bits	CTR DRBG	Generation: Derived from entropy, nonce and personalization string.	Input: N/A Output: N/A	N/A this is an internal state of DRBG	Plaintext in RAM	Zeroized by calling the zeroization service or by unloading the module.

Description/Usage	Type	Generation/Establishment	Input/Output	Key to Entity	Storage	Zeroization
DRBG Internal State Value of V (128 bits) and Key (AES 128, 192, 256 bits), entropy input (length dependent on security strength)	CTR_DRBG SP 800-90A CTR_DRBG (AES-128, 192, AES-256) with Derivation Function (AES-128, 192, AES-256) without Derivation Function	Generation: Derived from DRBG Seed.	Input: N/A Output: N/A	N/A this is an internal state of DRBG	Plaintext in RAM	Zeroized by calling the zeroization service or by unloading the module.
DRBG Seed Seed length: 440 or 888 bits	HMAC DRBG	Generation: Derived from entropy, nonce and personalization string.	Input: N/A Output: N/A	N/A this is an internal state of DRBG	The Module stores DRBG state values for the lifetime of the DRBG instance plaintext in RAM.	Zeroized by calling the zeroization service or by unloading the module.
DRBG internal state Value of V (160/224/256/384/512 bits) and Key (160/224/256/384/512 bits)	HMAC DRBG HMAC-SHA-1 HMAC-SHA-224 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	Generation: Derived from DRBG Seed.	Input: N/A Output: N/A	N/A this is an internal state of DRBG	The Module stores DRBG state values for the lifetime of the DRBG instance plaintext in RAM.	Zeroized by calling the zeroization service or by unloading the module.

Description/Usage	Type	Generation/Establishment	Input/Output	Key to Entity	Storage	Zeroization
DRBG Seed Seed length: 440 or 888 bits	Hash DRBG	Generation: Derived from entropy, nonce and personalization string.	Input: N/A Output: N/A	N/A this is an internal state of DRBG	The Module stores DRBG state values for the lifetime of the DRBG instance plaintext in RAM.	Zeroized by calling the zeroization service or by unloading the module.
DRBG internal state Value of V (440/888 bits) and C (440/888 bits)	Hash DRBG SHA-1 SHA-224 SHA-256 SHA-384 SHA-512	Generation: Derived from DRBG Seed.	Input: N/A Output: N/A	N/A this is an internal state of DRBG	The Module stores DRBG state values for the lifetime of the DRBG instance plaintext in RAM.	Zeroized by calling the zeroization service or by unloading the module.
EC DH Private	EC DH B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521		Input: passed in by the calling application on the stack. Output: in plaintext to the calling application	calling application user/CO	Plaintext in RAM, and Stored on Disk (responsibility of the calling application)	Zeroized by calling the zeroization service or by unloading the module.

Description/Usage	Type	Generation/Establishment	Input/Output	Key to Entity	Storage	Zeroization
ECC-CDH Primitive Computation (in support of 56ARev3 shared secret computation)	ECC CDH B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521	ECC-CDH Primitive Computation (in support of 56ARev3 shared secret computation)	Input: N/A Output: in plaintext to the calling application	calling application user/CO	Plaintext in RAM	Zeroized by unloading the module
HMAC Keys Keys used for: HMAC-SHA1: 112-bit min key HMAC-SHA224: 128-bit min key HMAC-SHA256: 128-bit min key HMAC-SHA384: 192-bit min key HMAC-SHA512: 256-bit min key	HMAC keys	SP 800-90A CTR_DRBG; As per SP 800-133r2 Section 4. Generated by DRBG as per the “Direct Generation” of Symmetric Keys which is an approved key generation method. Stored in RAM, until the zeroization function is called or the application is unloaded.	Input: in plaintext by the calling application Output: N/A	calling application user/CO	Plaintext in RAM	Zeroized by calling the zeroization service or by unloading the module.

Description/Usage	Type	Generation/Establishment	Input/Output	Key to Entity	Storage	Zeroization
RSA SGK Private Key Key sizes are 2048 to 4096 bits for Digital Signature.	RSA Private Key	As per SP 800-133r2, Section 4, generated by DRBG as per the “Direct Generation” of the seeds used for Asymmetric Keys which is an approved key generation method. Keys are generated and output to the calling application. Keys are stored in RAM and/or Disk. Keys will remain in RAM until the zeroization function is called or the application is unloaded. Persistent storage is the responsibility of the calling application.	Input: N/A Output: in plaintext to the calling application	calling application user/CO	Plaintext in RAM, and Stored on Disk (responsibility of the calling application)	Zeroized by calling the zeroization service or by unloading the module.

Description/Usage	Type	Generation/Establishment	Input/Output	Key to Entity	Storage	Zeroization
RSA KDK Private Key Key sizes are 2048 to 3072 bits for Key wrapping.	RSA Private Key	As per SP 800-133r2, Section 4, generated by DRBG as per the "Direct Generation" of the seeds used for Asymmetric Keys which is an approved key generation method. Keys are generated and output to the calling application. Keys are stored in RAM and/or Disk. Keys will remain in RAM until the zeroization function is called or the application is unloaded. Persistent storage is the responsibility of the calling application.	Input: N/A Output: in plaintext to the calling application	calling application user/CO	Plaintext in RAM, and Stored on Disk (responsibility of the calling application)	Zeroized by calling the zeroization service or by unloading the module.
TLS 1.2 Master Secret	Length: 48 bytes	Derived using TLS 1.2 KDF	Input: N/A Output: N/A	TLS 1.2 KDF Process	Plaintext in RAM	Zeroized by unloading the module
TLS 1.2 Secret (Pre-Master)	Secret (48 byte)		Input: calling application Output: N/A	TLS 1.2 KDF Process	Plaintext in RAM	Zeroized by unloading the module
TLS 1.2 KDF Internal state	TLS 1.2	Established using SP 800-135 TLS 1.2 KDF	Input: N/A Output: N/A	TLS 1.2 KDF Process	Plaintext in RAM during the lifetime of the KDF process	Zeroized by unloading the module

Description/Usage	Type	Generation/Establishment	Input/Output	Key to Entity	Storage	Zeroization
TLS 1.2 KDF Derived key material	TLS 1.2	Established using SP 800-135 TLS 1.2 KDF	Input: N/A Output: in plaintext to the calling application	TLS 1.2 KDF Process	Plaintext in RAM	Zeroized by unloading the module
IKEv1 Pre-shared Key	Pre-shared key (8-224 bit)		Input: calling application Output: N/A	IKEv1 KDF Process	Plaintext in RAM	Zeroized by unloading the module
IKEv1 Diffie-Hellman Shared Secret	DH 224, 2048, 8192		Input: passed in by the calling application on the stack. Output: N/A	IKEv1 KDF Process	Plaintext in RAM	Zeroized by unloading the module
IKEv1 KDF Internal State	IKEv1 Using SHA1, SHA224, SHA256, SHA384, SHA512	Established using SP 800-135 IKEv1 KDF	Input: N/A Output: N/A	IKEv1 KDF Process	Plaintext in RAM during the lifetime of the KDF process	Zeroized by unloading the module
IKEv1 KDF Derived key material	IKEv1 AES-CBC (128, 256 bits) and HMAC keys using SHA1, SHA224, SHA256, SHA384, SHA512	AES key and HMAC key are derived using IKEv1 KDF.	Input: N/A Output: in plaintext to the calling application	IKEv1 KDF Process	Plaintext in RAM	Zeroized by unloading the module
SSH Shared Secret	DH 2048		Input: in plaintext by the calling application on the stack. Output: N/A	SSH KDF Process	Plaintext in RAM	Zeroized by unloading the module

Description/Usage	Type	Generation/Establishment	Input/Output	Key to Entity	Storage	Zeroization
SSH KDF Internal state	SSH	Established using SP 800-135 SSH KDF	Input: N/A Output: N/A	SSH KDF Process	Plaintext in RAM	Zeroized by unloading the module
SSH KDF Derived key material	SSH	Established using SP 800-135 SSH KDF	Input: N/A Output: in plaintext to the calling application	SSH KDF Process	Plaintext in RAM	Zeroized by unloading the module
SNMP Password	Password (64-128 bits)		Input: in plaintext by the calling application Output: N/A	SNMP KDF Process	Plaintext in RAM	Zeroized by unloading the module
SNMP KDF Internal state	SNMP	Established using SP 800-135 SNMP KDF	Input: N/A Output: N/A	SNMP KDF Process	Plaintext in RAM during the lifetime of the KDF process	Zeroized by unloading the module
SNMP KDF Derived key material	SNMP	Established using SP 800-135 SNMP KDF	Input: N/A Output: in plaintext to the calling application	SNMP KDF Process	Plaintext in RAM	Zeroized by unloading the module

Note: As per SP 800-133r2, section 4, keys identified as being “Generated internally by calling FIPS Approved DRBG", the generated seed used in the asymmetric key generation is an unmodified output from the DRBG.

The table below describes Public keys used by the module.

Table 8 Public Keys

Description/Usage	Type	Generation	Input/Output	Key to Entity	Storage	Zeroization
EC DH Public Key	EC DH B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521		Input: passed in by the calling application on the stack. Output: N/A	calling application user/CO	Plaintext in RAM, and Stored on Disk (responsibility of the calling application)	Zeroized by calling the zeroization service or by unloading the module.
RSA SVK Public Key Key sizes are 1024 to 4096 bits for Digital Signature.	RSA Public Key	As per SP 800-133r2, Section 4, generated by DRBG as per the "Direct Generation" of the seeds used for Asymmetric Keys which is an approved key generation method. Keys are generated and output to the calling application. Keys are stored in RAM and/or Disk. Keys will remain in RAM until the zeroization function is called or the application is unloaded. Persistent storage is the responsibility of the calling application.	Input: N/A Output: in plaintext to the calling application	calling application user/CO	Plaintext in RAM, and Stored on Disk (responsibility of the calling application)	Zeroized by calling the zeroization service or by unloading the module.

Description/Usage	Type	Generation	Input/Output	Key to Entity	Storage	Zeroization
RSA KEK Public Key Key sizes are 2048 to 3072 bits for Key wrapping.	RSA Public Key	As per SP 800-133r2, Section 4, generated by DRBG as per the "Direct Generation" of the seeds used for Asymmetric Keys which is an approved key generation method. Keys are generated and output to the calling application. Keys are stored in RAM and/or Disk. Keys will remain in RAM until the zeroization function is called or the application is unloaded. Persistent storage is the responsibility of the calling application.	Input: N/A Output: in plaintext to the calling application	calling application user/CO	Plaintext in RAM, and Stored on Disk (responsibility of the calling application)	Zeroized by calling the zeroization service or by unloading the module.

6 Roles, Services and Authentication

6.1 Roles and Services

The Module implements the User and Crypto Officer roles in the FIPS mode of operation but does not implement authentication for those roles. Only one role may be active at a time; the Module does not allow concurrent operators. The User or Crypto Officer roles are implicitly assumed (as defined by guidance). Both roles have access to all the services provided by the Module. The Crypto Officer (CO) configures the operational environment for the module. The User is the calling application and consumes the cryptographic services provided by the module. After the module is instantiated, the user (application) invokes FIPS mode of operation.

All services implemented by the Module are listed below, along with a description of service Critical Security Parameters (CSP) access.

Table 9 Roles and Services in FIPS Mode of Operation

Service	Corresponding Roles	Description
Initialize	Crypto Officer and User	Module initialization. Does not access CSPs.
Self-test	Crypto Officer and User	Perform self-tests (FIPS_selftest). Does not access CSPs.
Zeroize	Crypto Officer and User	For a given DRBG context, overwrites DRBG CSPs (Hash_DRBG CSPs, HMAC_DRBG CSPs, CTR_DRBG CSPs.) All other services automatically overwrite CSPs stored in allocated memory. Stack cleanup is the responsibility of the calling application.
Show status	Crypto Officer and User	Functions that provide module status information: FIPS Mode Does not access CSPs.
Random number generation	Crypto Officer and User	Used for random number and symmetric key generation. • Seed or reseed a DRBG instance • Determine security strength of an RNG or DRBG instance • Obtain random data Uses and updates Hash_DRBG CSPs, HMAC_DRBG CSPs, CTR_DRBG CSPs

Service	Corresponding Roles	Description
Asymmetric key generation	Crypto Officer and User	Used to generate RSA keys. The SP800-90A DRBG supports an entropy strength of 256 bits
Symmetric encrypt/decrypt	Crypto Officer and User	Used to encrypt or decrypt data. Executes using AES (passed in by the calling process).
Symmetric digest	Crypto Officer and User	Used to generate or verify data integrity with AES-GCM. Executes using AES-GCM (passed in by the calling process).
Message digest	Crypto Officer and User	Used to generate a SHA-1 or SHA-2 message digest. Does not access CSPs.
Keyed Hash	Crypto Officer and User	Used to generate or verify data integrity with HMAC. Executes using HMAC Key (passed in by the calling process)
Asymmetric Encrypt/Decrypt [Note 1]	Crypto Officer and User	Used to encrypt or decrypt a key value on behalf of the calling process (does not establish keys into the module). Executes using RSA KDK, RSA KEK (passed in by the calling process)
Key agreement	Crypto Officer and User	Used to perform key agreement primitives on behalf of the calling process (does not establish keys into the module). Executes using EC DH Private, EC DH Public (passed in by the calling process).
Digital signature	Crypto Officer and User	Used to generate or verify RSA digital signature. Executes using RSA SGK and RSA SVK (passed in by the calling process).
Key Derivation	Crypto Officer and User	Used to derive key material for TLS 1.2, IKEv1, SSH and SNMP
Utility	Crypto Officer and User	Miscellaneous helper functions. Does not access CSPs.

Note 1: "Asymmetric Encrypt/Decrypt" can refer to a) moving keys in and out of the module or b) the use of keys by an external application. The latter definition is the one that applies to the Silver Peak ECOS Cryptographic library.

6.2 Authentication

The module does not implement authentication mechanisms.

Table 10 Roles and Required Identification and Authentication

Role	Type of Authentication	Authentication Data
User Role	N/A	N/A
Crypto-Officer Role	N/A	N/A

Table 11 Strengths of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
N/A	N/A
N/A	N/A

7 Operational Environment

The module runs on a General Purpose Computer (GPC) as a modifiable operational environment. The tested operating systems segregate user processes into separate process spaces. Each process space is logically separated from all other processes by the operating system software and hardware. The Module functions entirely within the process space of the calling application, and implicitly satisfies the FIPS 140-2 requirement for a single user mode of operation.

As per Table 2 in section 2, Tested Configurations, the operational environment includes operation with AES-NI enabled and with AES-NI disabled.

NOTE: The CMVP allows porting of this cryptographic module from the operational environment specified on the validation certificate to an operational environment which was not included as part of the validation testing as long as the porting rules of FIPS 140-2 Implementation Guidance G.5 are followed. As per FIPS 140-2 Implementation Guidance G.5, no claim can be made as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment which is not listed above in Table 2.

The cryptographic module is also supported on the following operational environments for which operational testing and algorithm testing was not performed.

7.1 All Silver Peak EdgeConnect hardware appliances

The following EdgeConnect hardware appliance configurations are vendor affirmed.

Table 12 Vendor Affirmed EdgeConnect Hardware Appliances

EdgeConnect HW Appliance (Note 1)	Operating System	Processor
EC-US	Fedora Core 6 (2.6.38 Kernel)[1] and Yocto 2.7.3 Warrior (4.19.87 Kernel)[2]	Intel® Atom™ CPU E3825@ 1.33 GHz
EC-XS	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Atom™ C2358 (Rangely), 1.7 GHz
EC-XS (2020)	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Atom™ C3558 (Denverton), 2.20 GHz
EC-S	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® CPU E3-1268L v3, 2.30GHz
EC-S-P	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® D-2123IT
EC-M	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® CPU E3-1270 v5, 3.60GHz

[1] ECOS version 8.1.9 uses Fedora Core 6 (2.6.38 Kernel) OS.

[2] ECOS version 9.1 uses Yocto 2.7.3 Warrior (4.19.87 Kernel) OS.

EC-M-P	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® CPU E-2176G 3.7GHz
EC-M-P	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® CPU E3-1270 v5, 3.60GHz
EC-M-H	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® D-2163IT,12C/24T (Skylake D), 2.10GHz
EC-L, EC-L-NM	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® CPU E5-2650 v3, 2.30GHz
EC-L-P, EC-L-P-NM	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® Gold 5118, 2.30 GHz
EC-L-H	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon-Gold 5218, 2.3GHz
EC-XL, EC-XL-NM	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® CPU E5-2680 v3, 2.50GHz
EC-XL-P, EC-XL-P-NM (10G)	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® Gold 6126, 2.60 GHz
EC-XL-P, EC-XL-P-NM (25G)	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon® Gold 6126, 2.60 GHz
EC-XL-H	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Intel® Xeon-Gold 5218, 2.3GHz

Note 1: All HW appliance include the -SP (Service Provider) models.

7.2 All Silver Peak EdgeConnect Virtual appliances

The following EdgeConnect virtual appliance configurations are vendor affirmed.

Table 13 Vendor Affirmed EdgeConnect Virtual Appliances

EdgeConnect Virtual Appliance	Operating System	Hypervisor
EC-V	Fedora Core 6 (2.6.38 Kernel)[3] and Yocto 2.7.3 Warrior (4.19.87 Kernel)[4]	VMware ESXi/ESX 6.7
EC-V	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	VMware ESXi/ESX 7.0
EC-V	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Red Hat KVM 8.x
EC-V	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	KVM, QEMU 4.x
EC-V	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Microsoft Hyper V 10.0
EC-V	Fedora Core 6 (2.6.38 Kernel) and Yocto 2.7.3 Warrior (4.19.87 Kernel)	Citrix Xen Server 8.1.0

[3] ECOS version 8.1.9 uses Fedora Core 6 (2.6.38 Kernel) OS.

[4] ECOS version 9.1 uses Yocto 2.7.3 Warrior (4.19.87 Kernel) OS.

8 Self-Tests

The module performs the following power-up and conditional self-tests after each reboot. Upon failure of either a power-up or conditional self-test, the module returns an error status and halts its operation. For all tests listed below, requirements of IG 9.10 have been met.

Upon successful completion of the power-on self-tests, the module displays the results to the console.

- `FIPS OpenSSL Power On Self Test Succeeded`
- `FIPS system files integrity check: OK`

Confirm self-tests completed by checking the messages and associated times on the console.

In the event of a KATs failure, the appliance logs different messages on the console, depending on the error.

- `FIPS OpenSSL Power-on Self-tests FAILED`
- `FIPS system files integrity check: FAILED`

Conditional self-test failure log message:

- `<conditional self-test name>......Failed!`

Table 14 Self-Tests

Algorithm	Test	Power-up/Conditional Self-test
AES	Known Answer Test (KAT) using ECB mode (encryption/decryption) AES-128 CBC mode (encryption/decryption) AES-128	Power-up self-test
AES-GCM	KAT using 256 key length	Power-up self-test
SHS	KAT using SHA1 (Note 1)	Power-up self-test
HMAC-SHA1	Software Integrity Test	Power-up self-test
HMAC	KAT using SHA1, SHA224, SHA256, SHA384 and SHA512	Power-up self-test
KDF	KAT for TLS 1.2 KAT for IKEv1 KAT for SSH KAT for SNMP	Power-up self-test
SP800-90A DRBG	KAT: CTR_DRBG HASH_DRBG HMAC_DRBG	Power-up self-test
	Continuous Random Number Generator Test (Note 2)	Conditional Self-test

Algorithm	Test	Power-up/Conditional Self-test
CRNGT	Continuous Random Number Generator test	Conditional Self-test
RSA	KAT using 2048 bit key, SHA-256 (sign / verify) KAT using 2048 bit key, SHA-256 (encryption / decryption)	Power-up self-test
	Pairwise Consistency Test (sign / verify) Pairwise Consistency Test (encryption / decryption)	Conditional Self-test
ECC CDH	Primitive "Z" Computation KAT, P-224	Power-up Self-test

Note 1: SHA1 has Known Answer Test (KAT). SHA224, SHA256, SHA384, and SHA512 are tested as part of HMAC.

Note 2: The module performs DRBG health tests as defined in Section 11.3 of SP800-90A, and continuous number generator test (CRNGT) to ensure that consecutive random numbers do not repeat.

9 Physical Security

The module is software only and does not have any physical security mechanisms.

Table 15 Inspection/Testing of Physical Security Mechanisms

Physical Security Mechanisms	Recommended Frequency of Inspection/Test	Inspection/Test Guidance Details
N/A	N/A	N/A

10 Mitigation of Other Attacks

The module provides no additional mitigation of other attacks.

Table 16 Mitigation of Other Attacks

Other Attacks	Mitigation Mechanism	Specific Limitations
N/A	N/A	N/A

11 Glossary and Definitions

The cryptographic module shall be a set of hardware, software, firmware, or some combination thereof that implements cryptographic functions or processes, including cryptographic algorithms and, optionally, key generation, and is contained within a defined cryptographic boundary.

Table 17 Glossary and Definitions

Abbreviation	Meaning
ACVP	Automated Cryptographic Validation Program
AES	Advanced Encryption Standard, as specified in [FIPS 197]
AES-GCM	AES with Galois/Counter Mode
ANSI	American National Standards Institute
CAVP	Cryptographic Algorithm Validation Program
CAVS	Cryptographic Algorithm Validation System
CBC	Cipher Block Chaining
CC	Common Criteria
CCM	Counter with Cipher Block Chaining-Message Authentication Code
Cert	Certificate
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CTR	Counter
DES	Data Encryption Standard
DRBG	Deterministic Random Bit Generator
EC DH	Elliptic Curve Diffie-Hellman (Algorithm)
ECC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman (NIST SP 800-56Ar3)
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard
FSM	Finite State Model
GCM	Galois Counter Mode (GCM) and GMAC Algorithm
GPC	General Purpose Computer
HMAC	Keyed-Hash Message Authentication Code, as specified in [FIPS 198]
IG	Implementation Guidance
IUT	Implementation Under Test
IV	Initialization Vector
KAS	Key Agreement Schemes and Key Confirmation (NIST SP 800-56Ar3)

MAC	Message Authentication Code
MD5	Message Digest 5
NIST	National Institute of Standards and Technology
NRBG	Non-deterministic Random Bit Generator
OS	Operating System
PKCS	Public Key Cryptography Standard
RNG	Random Number Generator
RSA	Rivest Shamir Adleman Cryptographic System (FIPS 186-4)
RSA	Reversible Digital Signature Algorithm (FIPS186-2 and FIPS186-3 RSA)
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SP	NIST Special Publication
SSH	Secure Shell
TDEA	Triple Data Encryption Algorithm, as specified in [SP 800-67]
TDES	Triple Data Encryption Standard
TID	Tracking Identification Number
TLS	Transport Layer Security

12 References

Table 18 References

Reference	Specification
[FIPS 140-2]	Security Requirements for Cryptographic modules, May 25, 2001
[FIPS 180-4]	Secure Hash Standard (SHS)
[FIPS 186-2/4]	Digital Signature Standard
[FIPS 197]	Advanced Encryption Standard
[FIPS 198-1]	The Keyed-Hash Message Authentication Code (HMAC)
[PKCS#1 v2.1]	RSA Cryptography Standard
[PKCS#5]	Password-Based Cryptography Standard
[PKCS#12]	Personal Information Exchange Syntax Standard
[SP 800-38A]	Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode
[SP 800-38F]	Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping
[SP 800-56Ar3]	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography
[SP 800-56B]	Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography
[SP 800-56C]	Recommendation for Key Derivation through Extraction-then-Expansion
[SP 800-67R1]	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
[SP 800-90A]	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
[SP 800-133r2]	Recommendation for Cryptographic Key Generation
[SP 800-135]	Recommendation for Existing Application –Specific Key Derivation Functions