Amazon Linux 2 Libreswan Cryptographic Module

Module Version 1.0

FIPS 140-2 Non-Proprietary Security Policy

Document Version 1.2

Last update: 2020-April-21

Prepared by:

atsec information security corporation

9130 Jollyville Road, Suite 260

Austin, TX 78759

www.atsec.com

Table of Contents

List of Tables

List of Figures

Copyrights and Trademarks

Amazon is a registered trademark of Amazon Web Services, Inc. or its affiliates.

1 Introduction

This document is the non-proprietary FIPS 140-2 Security Policy for version 1.0 of the Amazon Linux 2 Libreswan Cryptographic Module. It contains the security rules under which the module must be operated and describes how this module meets the requirements as specified in FIPS 140-2 (Federal Information Processing Standards Publication 140-2) for a Security Level 1 module.

1.1 Purpose of the Security Policy

There are three major reasons that a security policy is needed:

- It is required for FIPS 140 2 validation,
- It allows individuals and organizations to determine whether a cryptographic module, as implemented, satisfies the stated security policy, and
- It describes the capabilities, protection and access rights provided by the cryptographic module, allowing individuals and organizations to determine whether it will meet their security requirements.

1.2 Target Audience

This document is part of the package of documents that are submitted for FIPS 140 2 conformance validation of the module. It is intended for the following audience:

- Developers.
- FIPS 140-2 testing lab.
- The Cryptographic Module Validation Program (CMVP).
- Customers using or considering integration of Amazon Linux 2 Libreswan Cryptographic Module.

2 Cryptographic Module Specification

2.1 Module Overview

The Amazon Linux 2 Libreswan Cryptographic Module (hereafter referred to as the “module”) is a framework for providing cryptographic services to other network entities implementing the IKEv1 and IKEv2 protocols.

The module uses the Amazon Linux 2 NSS Cryptographic Module as a bound module (also referred to as “the bound NSS module”), which provides the underlying cryptographic algorithms, and the Amazon Linux 2 OpenSSL Cryptographic Module as a bound module (also referred to as “the bound OpenSSL module”), which provides the algorithm for integrity check.

2.2 FIPS 140-2 Validation Scope

Table 1 shows the security level claimed for each of the eleven sections that comprise the FIPS 140-2 standard.

Table 1: FIPS 140-2 Security Requirements.

Security Requirements Section		Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles and Services and Authentication	1
4	Finite State Machine 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
Overall Level		1

2.3 Definition of the Cryptographic Module

The Amazon Linux 2 Libreswan Cryptographic Module is defined as a Multi-chip Standalone software module per the requirements within FIPS 140-2. The logical cryptographic boundary of the module consists of the application, library files and their integrity test HMAC files as listed here:

- /usr/bin/fipscheck
- /usr/libexec/ipsec/pluto
- /usr/lib64/libfipscheck.so.1.2.1
- /usr/lib64/fipscheck/fipscheck.hmac
- /usr/lib64/fipscheck/libfipscheck.so.1.2.1.hmac
- /usr/lib64/fipscheck/pluto.hmac

The module is delivered through the Amazon Linux 2 yum core repository (ID amz2-core/2/x86_64) from the following RPMs which are need to be installed on the system.

- Pluto IKE Daemon and its HMAC file provided with package libreswan-3.23-5.amzn2.x86_64.
- fipscheck library with its HMAC file and fipscheck application with its HMAC file provided with packages fipscheck-1.4.1-6.amzn2.0.2.x86_64 and fipscheck-lib-1.4.1-6.amzn2.0.2.x86_64.

The following components which act as bound modules need to be installed for the Libreswan Module to operate.

- The bound Amazon Linux 2 OpenSSL Cryptographic Module with FIPS certificate #3553.
- The bound Amazon Linux 2 NSS Cryptographic Module with FIPS certificate #3646.

Figure 1 shows the logical block diagram of the module executing in memory on the host system. The logical cryptographic boundary is indicated with a dashed colored box.

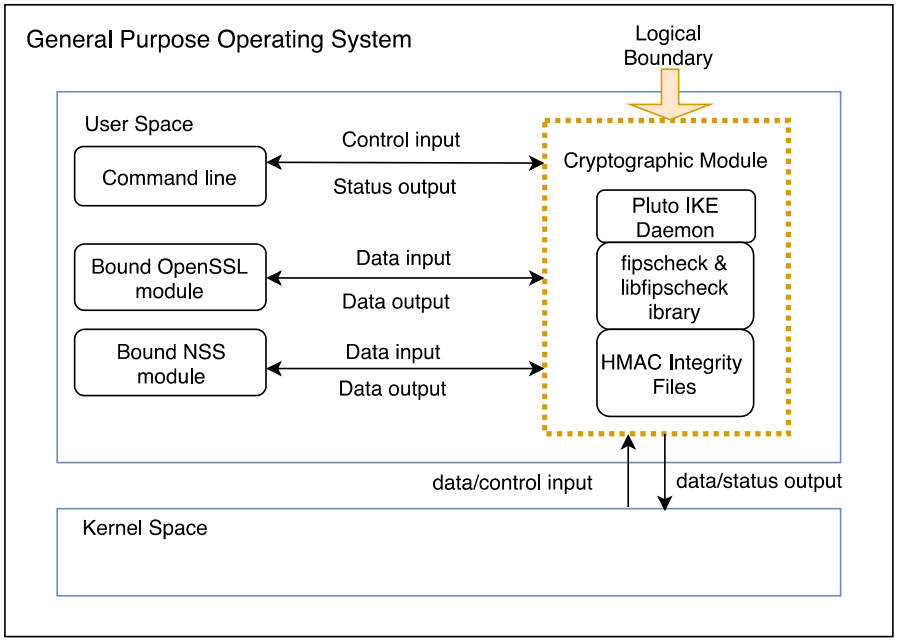

Figure 1: Logical cryptographic boundary.

2.4 Definition of the Physical Cryptographic Boundary

The physical cryptographic boundary of the module is defined as the hard enclosure of the host system on which the module runs. Figure 2 depicts the hardware block diagram. The physical hard enclosure is indicated by the dashed colored line. No components are excluded from the requirements of FIPS 140-2.

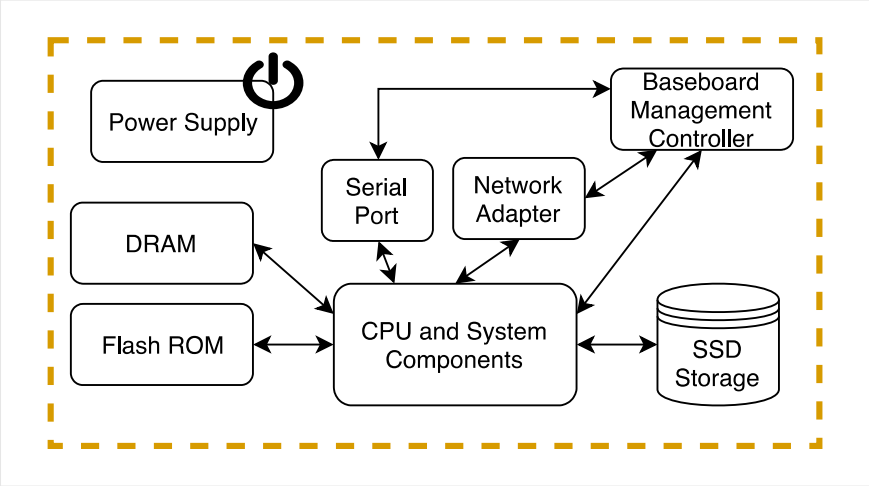

Figure 2: Hardware block diagram.

2.5 Tested Environments

The module was tested on the environments/platforms listed in Table 2. The tested operational environment was controlled and the laboratory had full and exclusive access to the environment and module during the testing procedures.

Table 2: Tested operational environments.

Operating System	Processor	Hardware
Amazon Linux 2	Intel Xeon E5 (Broadwell) x86_64bit with PAA (i.e., AES-NI)	Amazon EC2 i3.metal 512 GiB system memory 13.6 TiB SSD storage + 8 GiB SSD boot disk 25 Gbps Elastic Network Adapter
Amazon Linux 2	Intel Xeon E5 (Broadwell) x86_64bit without PAA (i.e., AES-NI)	Amazon EC2 i3.metal 512 GiB system memory 13.6 TiB SSD storage + 8 GiB SSD boot disk 25 Gbps Elastic Network Adapter

2.6 Modes of Operation

The module supports two modes of operation.

- In "**FIPS mode**" (the Approved mode of operation), only approved or allowed security functions with sufficient security strength are offered by the module.
- In "**non-FIPS mode**" (the non-Approved mode of operation), non-approved security functions are offered by the module.

The module enters the operational mode after Power-On Self-Tests (POST) succeed. Once the module is operational, the mode of operation is implicitly assumed depending on the security function invoked and the security strength[1] of the cryptographic keys chosen for the service.

If the POST fails (Section 9), the module goes into the error state. The status of the module can be determined by the availability of the module. If the module is available, then it has passed all self-tests. If the module is unavailable, it is because one of the self-tests failed, and the module has transitioned to the error state.

Keys and Critical Security Parameters (CSPs) used or stored in FIPS mode shall not be used in non-FIPS mode, and vice versa.

[1] See Section 5.6.1 in [SP800-57] for a definition of “security strength”.

3 Module Ports and Interfaces

As a Software module, the module does not have physical ports. Thus, the physical ports within the physical boundary are interpreted to be the physical ports of the hardware platform on which the module runs and are directed through the interfaces provided by the module. Table 3 summarizes the module's interfaces.

Table 3: Ports and interfaces.

FIPS 140-2 Interface	Physical Port	Module Interfaces
Data Input	Ethernet port	IKE Network Port/Protocol, NSS Key Database file stored in /etc/ipsec.d/
Data Output	Ethernet Port	IKE Network Port/Protocol, Linux Kernel (netlink/XFRM Interface)
Control Input	Management Ethernet Port, USB for Keyboard/Mouse, Serial Port	IKE Network Port/Protocol, Configuration Files (/etc/ipsec.conf, /etc/ipsec.d/, /etc/ipsec.secrets), Linux Kernel (netlink/XFRM Interface), command line
Status Output	Management Ethernet Port, Serial Port	Log File, IKE Network Port/Protocol
Power Input	PC power supply	N/A

4 Roles, Services and Authentication

4.1 Roles

The module supports the following roles:

- **User role**: performs key derivation and IKE negotiation, manages Pluto IKE daemon, self-tests, show status and zeroization services. This role is assumed by the entity using the module.
- **Crypto Officer role**: performs module installation and configuration. This role is assumed by the entity installing the module.

The User and Crypto Officer roles are implicitly assumed depending on the service requested.

4.2 Services

Table 4 and Table 5 depict all services. The tables use the following convention when specifying the access permissions that the module has for each CSP or key. Access types are indicated by the abbreviation within parentheses.

- **Create (C)**: the calling application can create a new CSP.
- **Read (R)**: the calling application can read the CSP.
- **Update (U)**: the calling application can write a new value to the CSP.
- **Zeroize (Z)**: the calling application can zeroize the CSP.
- **N/A**: the calling application does not access any CSP or key during its operation.

For the “Role” column, U indicates the User role, and CO indicates the Crypto Officer role. An X marks which role has access to that service.

4.2.1 Services in the FIPS-Approved Mode of Operation

Table 4 provides a full description of FIPS Approved services and the non-Approved but Allowed services provided by the module in the FIPS-approved mode of operation and lists the roles allowed to invoke each service.

Table 4: Services in the FIPS-approved mode of operation.

Service	Service Description and Algorithms	Role		Keys and CSPs	Access Types
		U	CO		
Module Installation and configuration	Installation and configuration of the cryptographic module		X	RSA Private Key, pre-shared key	C, R, U
Manage Pluto IKE Daemon	Manage Pluto IKE daemon like start and stop activities and destroy keys	X		All Keys and CSPs	R
Negotiate IKE to Establish Security Associations (SA’s)	Negotiate key agreement using IKE to establish security associations	X		RSA, EC/Diffie-Hellman public/private keys, EC/Diffie-Hellman shared secret, IKE SA Tunnel Encryption Keys, IKE SA Tunnel Integrity Keys, IPsec	U

Service	Service Description and Algorithms	Role		Keys and CSPs	Access Types
		U	CO		
				SA encryption keys, IPsec SA Tunnel Integrity Keys	
Zeroize	Zeroize keys and CSP's	X		Keys and CSPs	Z
Self-Test	Perform on-demand self-tests	X		None	N/A
Show Status	Show status of the module	X		None	N/A

4.2.2 Services in the Non-FIPS-Approved Mode of Operation

Table 5 presents the services only available in non-FIPS-approved mode of operation.

Table 5: Services in the non-FIPS approved mode of operation.

Service	Service Description and Algorithms	Role		Keys and CSPs	Access Types
		U	CO		
Module Installation and configuration (non-approved keys)	Installation and configuration of the cryptographic module		X	RSA public/private key listed in Table 8.	C, R, U
Negotiate IKE to Establish Security Associations (SAs) (non-approved keys)	Negotiate key agreement using IKE to establish security associations	X		RSA, DH public/private key listed in Table 8.	C, R, U

4.3 Algorithms

The module implements the IKE KDF algorithm. The rest of the cryptographic algorithms are provided by the bound NSS module. The OpenSSL module only provides the integrity check algorithms. The cryptographic algorithms that are approved to be used in the FIPS mode of operation are tested and validated by CAVP. No parts of the IKE protocol have been tested by the CAVP, but for the key derivation function (KDF).

Table 6, Table 7 and Table 8 present the cryptographic algorithms in specific modes of operation. These tables include the CAVP certificates for different implementations, the algorithm name, respective standards, the available modes and key sizes wherein applicable, and usage. Information from certain columns may be applicable to more than one row.

4.3.1 FIPS-Approved

Table 6 lists the cryptographic algorithms that are approved to be used in the FIPS mode of operation.

Table 6: FIPS-approved cryptographic algorithms.

Algorithm	Standard	Mode	Key size	Use	CAVP Cert#
KDF	[SP800-135]	IKEv1, IKEv2	N/A	Key Derivation	#C807
Algorithms from the bound NSS Module					
AES	[FIPS197] [SP800-38A]	CBC, CTR	128, 192 and 256 bits	Data Encryption and Decryption	#C803 #C804
Triple-DES	[SP800-67] [SP800-38A]	CBC	192 bits	Data Encryption and Decryption	#C803
HMAC	[FIPS198-1]	SHA-1, SHA-256, SHA-384 SHA-512	112 bits or greater	Message Authentication Code	#C803
SHS	[FIPS180-4]	SHA-1, SHA-256, SHA-384, SHA-512	N/A	Message Digest	#C803
DRBG	[SP800-90A]	Hash_DRBG	n/a	Random Number Generation	#C803
DSA	[FIPS186-4]		L=2048, N=224; L=2048, N=256; L=3072, N=256	Key generation	#C803
ECDSA	[FIPS186-4]		P-256, P-384, P-521	Key generation	#C803
RSA	[FIPS 186-4]	SHA-224, SHA-256, SHA-384, SHA-512	2048,3072,4096	Signature Generation	#C803
		SHA-1, SHA-256, SHA-384, SHA-512	1024, 2048, 3072	Signature Verification	
KAS FFC Component	[SP800-56A]	FFC dhEphem scheme	2048 bits	Shared secret computation	#C803

Algorithm	Standard	Mode	Key size	Use	CAVP Cert#
KAS ECC Component	[SP800-56A]	ECC Ephemeral Unified scheme	P-256, P-384, P-521	Shared secret computation	#C803
Algorithms from the bound OpenSSL Module					
HMAC	[FIPS198-1]	SHA-256	112 bits or greater	Module Integrity	#C523 #C524, #C525 #C526

4.3.2 Non-Approved-but-Allowed

Table 7 lists the non-Approved-but-Allowed cryptographic algorithms provided by the bound NSS module that are allowed to be used in the FIPS mode of operation.

Table 7: Non-Approved-but-allowed cryptographic algorithms from bound NSS Module.

Algorithm	Usage
NDRNG	Used for seeding NIST SP 800-90A DRBG.
Diffie-Hellman	Shared secret computation with key sizes between 3072 and 8192 bits

The Libreswan and the bound NSS modules together provide the Diffie Hellman and EC Diffie Hellman key agreement. The Libreswan module only implements the KDF portion of the key agreement and the bound NSS module provides the shared secret computation.

- Diffie-Hellman with key sizes between 2048 and 8192 bits provides between 112 and 202 bits of encryption strength.
- EC Diffie-Hellman with P-256, P-384, P-521 curves provides between 128 and 256 bits of encryption strength.

4.3.3 Non-Approved

Table 8 lists the cryptographic algorithms that are not allowed to be used in the FIPS mode of operation. Use of any of these algorithms (and corresponding services in Table 5) will implicitly switch the module to the non-Approved mode.

Table 8: Non-FIPS approved cryptographic algorithms from bound NSS module.

Algorithm	Usage
AES GCM	Encryption/decryption
Diffie-Hellman shared secret computation	using 1024-bit key
RSA Signature generation	using keys less than 2048
SHA-1	Signature generation

4.4 Operator Authentication

The module does not support operator authentication mechanisms. The role of the operator is implicitly assumed based on the service requested.

5 Physical Security

The module is comprised of software only and thus this Security Policy does not claim any physical security.

6 Operational Environment

6.1 Applicability

The module operates in a modifiable operational environment per FIPS 140-2 Security Level 1 specifications. The module runs on the Amazon Linux 2 operating system executing on the hardware specified in Section 2.5.

6.2 Policy

The operating system is restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded by the operating system).

7 Cryptographic Key Management

Table 9 summarizes the keys and other CSPs that are used by the cryptographic services implemented in the module.

Table 9: Lifecycle of keys and other Critical Security Parameters (CSPs).

Name	Use	Generation	Entry and Output
IKE SA Tunnel Encryption Keys (AES, Triple-DES)	IKE session keys used for encryption and decryption.	Derived from shared secret using SP800-135 KDF.	Entry: N/A. Output: via API parameter to bound NSS module.
IKE SA Tunnel authentication key (HMAC)	IKE session key used for data authentication.	Derived from shared secret using SP800-135 KDF.	Entry: N/A. Output: via API parameter to bound NSS module.
IPsec SA Tunnel Encryption Keys (AES, Triple-DES)	IPsec session keys used for encryption and decryption.	Derived from shared secret using SP800-135 KDF.	Entry: N/A. Output: via API parameter to bound NSS module.
IPsec SA Tunnel authentication key (HMAC)	IPsec session key used for data authentication.	Derived from shared secret using SP800-135 KDF.	Entry: N/A. Output: via API parameter to bound NSS module.
Shared secret	Used to derive session keys.	Established by EC/Diffie-Hellman key agreement from bound NSS module	Entered via API input parameter from bound NSS module. No output.
RSA public/private key	Keys used for peer authentication.	keys are read from the key file	Entry: read from the /etc/ipsec.secrets file Output: N/A
Pre-shared Key	Pre-defined value provided to the module used to derive session keys for the IKE		
Diffie-Hellman public/private key	Used in Key agreement.	N/A (generated by the bound NSS module)	Entry via API input parameter from bound NSS module. Output via API input parameter to the bound NSS module.
EC Diffie-Hellman public/private key	Used in Key agreement.		

7.1 Random Number Generation and Key generation

The module does not implement any random number generator, nor does it provide key generation.

7.2 Key Derivation

The module only provides key derivation through the implementation of the SP 800-135 IKE KDF.

7.3 Key Entry/Output

The module does not support manual key entry. The keys can be entered into or output from the module electronically.

7.4 Key/CSP Storage

The module does not perform persistent storage of keys. The keys and CSPs are temporarily stored as plaintext in the RAM.

7.5 Key/CSP Zeroization

The destruction functions, overwrites the memory that is occupied by the key information with predefined value before it is deallocated.

8 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The test platforms listed in Table 2 have been tested and found to conform to the EMI/EMC requirements specified by 47 Code of Federal Regulations, FCC PART 15, Subpart B, Unintentional Radiators, Digital Devices, Class A (i.e., Business use). These devices are designed to provide reasonable protection against harmful interference when the devices are operated in a commercial environment.

9 Self-Tests

9.1 Power-Up Self-Tests

The module performs power-up self-tests (POSTs) automatically during initialization of the module. These POSTs ensure that the module is not corrupted. No operator intervention is necessary to run the POSTs. While the module is executing the POSTs, services are not available, and input and output are inhibited. The module is not available for use until successful completion of the POSTs.

The integrity check of the module (i.e. for all the components referenced in section 2.3) is performed by the fipscheck application using the HMAC-SHA-256 algorithm implemented by the bound Amazon Linux 2 OpenSSL Cryptographic Module. The pluto binary links with the library libfipscheck.so which is intended to execute fipscheck application to verify the integrity of the pluto binary using the HMAC-SHA-256 algorithm. Upon calling the FIPSCHECK_verify() function provided with libfipscheck.so, fipscheck is loaded and executed, and the following steps are performed:

1. OpenSSL, loaded by fipscheck, performs its own integrity check using the HMAC-SHA-256 algorithm.
2. fipscheck performs the integrity check of its own application file using the HMAC-SHA-256 algorithm provided by the OpenSSL Module.
3. fipscheck automatically verifies the integrity of libfipscheck.so before processing requests of calling applications.
4. The fipscheck application performs the integrity check of the Libreswan application file. The fipscheck computes the HMAC-SHA-256 checksum of the file and compares the computed value with the value stored inside the */usr/lib64/fipscheck/<application filename>.hmac* checksum file. The fipscheck application returns the appropriate exit value based on the comparison result: zero if the checksum is OK, or an error code otherwise.

If any of the above steps fail, an error code is returned and the Libreswan Module enters the Error state. In the Error state, all data output is inhibited and no cryptographic operation is allowed. The module needs to be reloaded to recover from the Error state. On successful completion of the tests, the module becomes operational and crypto services are then available. The Libreswan module uses the bound Amazon Linux 2 NSS Cryptographic Module which provides the underlying cryptographic algorithms. All the known answer tests are implemented by the bound NSS Module.

9.2 On-Demand self-tests

The module provides the Self-Test service to perform self-tests on demand. On demand self-tests can be invoked by powering-off and reloading the module. This service performs the same cryptographic algorithm tests executed during power-up. During the execution of the on-demand self-tests, cryptographic services are not available and no data output or input is possible.

10 Guidance

This section provides guidance for the Crypto Officer and the User to maintain proper use of the module per FIPS 140-2 requirements.

10.1 Crypto-Officer Guidance

The RPM files containing the FIPS validated module referenced in Section 2.3 must be installed according to this guidance.

As stated in the *Guidance* section of the Amazon Linux 2 NSS Cryptographic Module Security Policy, after configuring the operating environment to support FIPS, the file /proc/sys/crypto/fips_enabled will contain 1. If the file does not exist or does not contain “1”, the operating environment is not configured to support FIPS mode and the module will not operate as a FIPS validated module.

After performing the configuration described above, the Crypto Officer should proceed for module installation with the version of the RPM package listed in Section 2.3. The integrity of the RPM is automatically verified during the installation of the modules and the Crypto Officer shall not install the RPM file if the RPM tool indicates an integrity error.

10.1.1 Libreswan Configuration

With the operational environment setup as stated in the above section, the following restrictions are applicable.

- Configure Pluto as specified in ipsec.conf(5), and ipsec.secrets(5) man pages, as well as the file README.nss provided by the RPM package listed in section 2.3.
- To start and stop the module, use the (ipsec start or ipsec restart) command. This also performs the on-demand self test for the module.
- ikelifetime should not be larger than 1 hour.
- salifetime should not be larger than 1 hour.
- Aggressive mode should not be used.
- Only the FIPS 140-2 approved and allowed ciphers listed in section 3.1 shall be used in configuring the Pluto daemon. Use of non-approved cipher will put the module in the Non FIPS mode implicitly.
- The database for the cryptographic keys used by the Pluto Daemon must be initialized after it has been created as documented in the README.nss documentation with the following command, assuming that the database is stored in the directory /etc/ipsec.d/. “*modutil -fips true -dbdir /etc/ipsec.d*”

NOTE: Encryption and decryption of data is done implicitly when the kernel triggers Pluto to set up a new Security Association.

10.2 User Guidance

For the module, the mode of operation is implicitly assumed depending on the services/security functions invoked as stated in section 4.2 and the successive sections lists the available ciphers from the module. Any use of non-approved ciphers or non-Approved key sizes will result in the module entering the non-FIPS mode of operation.

- Starting the module using ipsec start or ipsec restart command performs the on-demand self test for the module.

- Stopping the module with ipsec stop will zeroize the ephemeral CSPs and keys
- To check module status, read the Pluto debug data using the ipsec_barf(8) tool.

10.3 Handling Self-Test Errors

The Libreswan self-tests consist of the software integrity test. If the integrity test fails, Libreswan enters the Error state. To recover from the Error state, the module must be restarted. If the failure persists, the module must be reinstalled. The bound OpenSSL and NSS modules' self-tests failures will prevent Libreswan from operating. See the Guidance section in the OpenSSL Security Policy and NSS Security Policy for instructions on handling the respective self-test failures.

11 Mitigation of Other Attacks

The module does not mitigate against attacks.

12 Acronyms, Terms and Abbreviations

Term	Definition
AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CMVP	Cryptographic Module Validation Program
CSE	Communications Security Establishment
CSP	Critical Security Parameter
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DRBG	Deterministic Random Bit Generator
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
HMAC	(Keyed) Hash Message Authentication Code
KAT	Known Answer Test
KDF	Key Derivation Function
NDRNG	Non-Deterministic Random Number generator
NIST	National Institute of Standards and Technology
PAA	Processor Algorithm Acceleration
POST	Power On Self Test
PR	Prediction Resistance
PSS	Probabilistic Signature Scheme
PUB	Publication
SHA	Secure Hash Algorithm
IKE	Internet Key Exchange

13 References

The FIPS 140-2 standard, and information on the CMVP, can be found at http://csrc.nist.gov/groups/STM/cmvp/index.html. More information describing the module can be found on the vendor web site at aws.amazon.com .

This Security Policy contains non-proprietary information. All other documentation submitted for FIPS 140-2 conformance testing and validation is proprietary and is releasable only under appropriate non-disclosure agreements.

Document	Author	Title
FIPS 140-2	NIST	FIPS 140-2: Security Requirements for Cryptographic Modules
FIPS IG	NIST	Implementation Guidance for FIPS 140-2 and the Cryptographic Module Validation Program
FIPS 140-2 Annex A	NIST	FIPS 140-2 Annex A: Approved Security Functions
FIPS 140-2 Annex B	NIST	FIPS 140-2 Annex B: Approved Protection Profiles
FIPS 140-2 Annex C	NIST	FIPS 140-2 Annex C: Approved Random Number Generators
FIPS 140-2 Annex D	NIST	FIPS 140-2 Annex D: Approved Key Establishment Techniques
DTR for FIPS 140-2	NIST	Derived Test Requirements (DTR) for FIPS 140-2, Security Requirements for Cryptographic Modules
NIST SP 800-67	NIST	Recommendation for the Triple Data Encryption Algorithm TDEA Block Cipher
FIPS PUB 197	NIST	Advanced Encryption Standard
FIPS PUB 198-1	NIST	The Keyed Hash Message Authentication Code (HMAC)
FIPS PUB 186-4	NIST	Digital Signature Standard (DSS)
FIPS PUB 180-4	NIST	Secure Hash Standard (SHS)
NIST SP 800-131A	NIST	Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes