WatchGuard Technologies, Inc.

WatchGuard Firebox M4800 and M5800

FIPS 140-3 Non-Proprietary Security Policy

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is a FIPS 140-3 non-proprietary Security Policy for WatchGuard Firebox M4800 and M5800, running firmware version 12.11 (hereinafter referred to as Module). This policy describes how the WatchGuard Firebox models (hereafter referred to as the 'module' or the 'Firebox module') meets the FIPS 140-3 security requirements and how to operate the module in an approved manner.

The following details how this module meets the security requirements of FIPS 140-3, SP 800-140 and ISO/IEC 19790 for a Security Level 2 Hardware cryptographic module.

The security requirements cover areas related to the design and implementation of a cryptographic module. These areas include cryptographic module specification; cryptographic module interfaces; roles, services, and authentication; software/firmware security; operational environment; physical security; non-invasive security; sensitive security parameter management; self-tests; life-cycle assurance; and mitigation of other attacks. The following table indicates the actual security levels for each area of the cryptographic module.

The WatchGuard Firebox appliances meet the overall requirements applicable to Level 2 security of FIPS 140-3.

1.2 Security Levels

Section	Title	Security Level
1	General	2
2	Cryptographic module specification	2
3	Cryptographic module interfaces	2
4	Roles, services, and authentication	3
5	Software/Firmware security	2
6	Operational environment	N/A
7	Physical security	2
8	Non-invasive security	N/A
9	Sensitive security parameter management	2
10	Self-tests	2
11	Life-cycle assurance	2
12	Mitigation of other attacks	N/A
	Overall Level	2

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Firebox module is a multi-chip standalone hardware cryptographic module built for enterprise-grade performance with blazing throughput and numerous connectivity options.

Advanced networking features include clustering, high availability (active/active), VLAN support, multi-WAN load balancing and enhanced VoIP security, plus inbound and outbound HTTPS inspection, to give the strong security enterprises need. And the Firebox appliances are completely configurable – turn on or off components and services to fit different network security deployment requirements. The module is operated in a limited operational environment**.**

Module Type: Hardware

Module Embodiment: MultiChipStand

Module Characteristics:

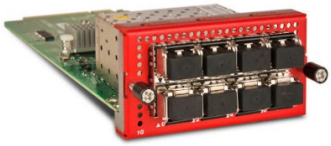
Cryptographic Boundary:

Module's cryptographic boundary is defined as the entire chassis unit's physical perimeter encompassing the "top," "front," "left," "right," "rear" and "bottom" surfaces of the case, and shown in the figures below and in Physical Security section.

Firebox M4800

Figure 1: M4800 Front View

8 x 1Gb Copper Module



8 x 1Gb SFP Module

4 x 10Gb SFP+ Module

2 x 40Gb QSFP+ Module

Figure 2: M4800 Rear View

Firebox M5800

Figure 3: M5800 Front View

8 x 1Gb Copper Module

8 x 1Gb SFP Module

4 x 10Gb SFP+ Module

2 x 40Gb QSFP+ Module

Figure 4: M5800 Rear View

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
M4800	M4800	12.11	Intel E-2176G	N/A
M5800	M5800	12.11	Intel 6230	N/A

Table 2: Tested Module Identification – Hardware

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

N/A for this module.

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

2.3 Excluded Components

N/A for this module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	The module always uses Approved Algorithms for its SSH and TLS services, and operates in Approved mode when the CO configures only Approved algorithms (and not legacy algs) for the IPSec/IKE service	Approved	Equivalent to the indicator of the requested service as defined in section 4.3
Non-Approved mode	Automatically entered whenever CO configures legacy algorithms for IPSec/IKE (MD5, DES, and Triple-DES) for interoperability with legacy system	Non-Approved	Equivalent to the indicator of the requested service as defined in section 4.4

Table 3: Modes List and Description

Mode Change Instructions and Status:

When the module starts up successfully, after passing all the pre-operational self-test and the cryptographic algorithms self-tests (CASTs), the module is capable of providing both Approved services (which utilize Approved algorithms) and non-Approved services/Algorithms (the Module allows configuration of a limited set of non-Approved algorithms [DES, Triple-DES, and MD5] which allow IPsec/IKE interoperability with legacy systems dependent on those algorithms).

The Module will not perform any non-Approved services/algorithms until the CO configures a new IPsec/IKE connection to use DES, Triple-DES, or MD5. After such configuration, the Module would then use the configured non-Approved algorithm as part of an IPsec connection with a peer device.

The CO can remove any IPsec/IKE connections configured to use legacy algorithms, and thus ensure that the module uses only approved Algorithms/services.

The module does not claim implementation of a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4397	-	SP 800-38A
AES-CTR	A4397	-	SP 800-38A
AES-GCM	A4397	-	SP 800-38D
Counter DRBG	A4397	-	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A4397	-	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A4397	-	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A4397	-	FIPS 186-5
HMAC-SHA-1	A4397	-	FIPS 198-1
HMAC-SHA2-256	A4397	-	FIPS 198-1
HMAC-SHA2-384	A4397	-	FIPS 198-1
HMAC-SHA2-512	A4397	-	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4397	-	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A4397	-	SP 800-56A Rev. 3
KDF IKEv1 (CVL)	A4397	-	SP 800-135 Rev. 1
KDF IKEv2 (CVL)	A4397	-	SP 800-135 Rev. 1
KDF SSH (CVL)	A4397	-	SP 800-135 Rev. 1
RSA KeyGen (FIPS186-5)	A4397	-	FIPS 186-5
RSA SigGen (FIPS186-5)	A4397	-	FIPS 186-5
RSA SigVer (FIPS186-5)	A4397	-	FIPS 186-5
Safe Primes Key Generation	A4397	-	SP 800-56A Rev. 3
SHA-1	A4397	-	FIPS 180-4
SHA2-256	A4397	-	FIPS 180-4
SHA2-384	A4397	-	FIPS 180-4
SHA2-512	A4397	-	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A4397	-	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A4397	-	SP 800-135 Rev. 1

Table 4: Approved Algorithms

- For TLS v1.2, the Module's AES-GCM implementation conforms to Implementation Guidance C.H scenario #1 following RFC 5288 for TLS. The Module is compatible with TLSv1.2 and provides support for the acceptable GCM cipher suites from SP 800-52 Rev1, Section 3.3.1. The keys for the client and server negotiated in the TLSv1.2 handshake process (client_write_key and server_write_key) are compared and the

Module aborts the session if the key values are identical. The operations of one of the two parties involved in the TLS key establishment scheme were performed entirely within the cryptographic boundary of the Module being validated. The counter portion of the IV is set by the Module within its cryptographic boundary. When the IV exhausts the maximum number of possible values for a given session key, the first party, client or server, to encounter this condition will trigger a handshake to establish a new encryption key. In case the Module's power is lost and then restored, a new key for use with the AES GCM encryption/decryption shall be established.

- For TLS v1.3, the Module offers the AES-GCM implementation and uses the context of Scenario #5 of FIPS 140-3 IG C.H. The protocol that provides this compliance is TLS 1.3, defined in RFC8446 of August 2018, using the cipher suites that explicitly select AES-GCM as the encryption/decryption cipher (Appendix B.4 of RFC8446). The Module supports acceptable AES-GCM cipher suites from Section 3.3.1 of SP800-52 Rev2. The Module implements, within its boundary, an IV generation unit for TLS 1.3 that keeps control of the 64-bit counter value within the AES-GCM IV. If the exhaustion condition is observed, the Module will return an error indication to the calling application, who will then need to either trigger a re-key of the session (i.e., a new key for AES-GCM), or terminate the connection.

- The Module uses RFC 7296 compliant IKEv2 to establish the shared secret SKEYSEED from which the AES GCM encryption keys are derived. Two keys established by IKEv2 for one security association (one key for encryption in each direction between the parties) are not identical and abort the session if they are. When the IV exhausts the maximum number of possible values for a given session key, the first party, client or server, to encounter this condition will trigger a handshake to establish a new encryption key. In case the Module's power is lost and then restored, a new key for use with the AES GCM encryption/decryption shall be established.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type:Asymmetric	N/A	The cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per sections 4 and 5 in SP800-133rev2 (vendor affirmed) and FIPS 140-3 IG D.H. A seed (i.e., the random value) used in asymmetric key generation is a direct output from SP800-90Arev1 CTR_DRBG (A4397)

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
MD5	Message digest
DES	Data encryption/decryption
Triple-DES	Data encryption/decryption

Table 6: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
KAS-ECC-KeyGen (SSHv2)	KAS-KeyGen	KAS ECC keygen used in SSHv2 service	Bit-strength Caveat:Provides between 128 and 256 bits encryption strength	Counter DRBG: (A4397) CKG: ()
KAS-FFC-KeyGen (SSHv2)	KAS-KeyGen	KAS FFC keygen used in SSHv2 service	Bit-strength Caveat:Provides 112 or 128 bits encryption strength	Counter DRBG: (A4397) Safe Primes Key Generation: (A4397) Safe Prime Groups: MODP-2048, MODP-3072 CKG: ()
KAS-ECC-KeyGen (TLSv1.2/v1.3)	KAS-KeyGen	KAS ECC keygen used in TLSv1.2/v1.3 service	Bit-strength Caveat:Provides between 128 and 256 bits encryption strength	Counter DRBG: (A4397) CKG: ()
KAS-FFC-KeyGen (TLSv1.2/v1.3)	KAS-KeyGen	KAS FFC keygen used in TLSv1.2/v1.3 service	Bit-strength Caveat:Provides 112 or 128 bits encryption strength	Counter DRBG: (A4397) Safe Primes Key Generation: (A4397) Safe Prime Groups: ffdhe2048, ffdhe3072 CKG: ()

Name	Type	Description	Properties	Algorithms
KAS-ECC-KeyGen (IKEv1/v2)	KAS-KeyGen	KAS ECC keygen used in IKEv1/v2 service	Bit-strength Caveat:Provides between 128 and 256 bits encryption strength	Counter DRBG: (A4397) CKG: ()
KAS-FFC-KeyGen (IKEv1/v2)	KAS-KeyGen	KAS FFC keygen used in IKEv1/v2 service	Bit-strength Caveat:Provides 112 or 128 bits encryption strength	Counter DRBG: (A4397) Safe Primes Key Generation: (A4397) Safe Prime Groups: ffdhe2048, ffdhe3072 CKG: ()
KAS-ECC (SSHv2)	KAS-Full	Full KAS-ECC key agreement used in SSHv2 service	Bit-strength Caveat:Provides between 128 and 256 bits encryption strength	KAS-ECC-SSC Sp800-56Ar3: (A4397) KDF SSH: (A4397)
KAS-FFC (SSHv2)	KAS-Full	Full KAS-FFC key agreement used in SSHv2 service	Bit-strength Caveat:Provides 112 or 128 bits encryption strength	KAS-FFC-SSC Sp800-56Ar3: (A4397) Domain Parameter Generation Method : MODP-2048, MODP-3072 KDF SSH: (A4397)
KAS-ECC (TLSv1.2/v1.3)	KAS-Full	Full KAS-ECC key agreement used in TLSv1.2/v1.3 service	Bit-strength Caveat:Provides between 128 and 256 bits encryption strength	TLS v1.2 KDF RFC7627: (A4397) TLS v1.3 KDF: (A4397) KAS-ECC-SSC Sp800-56Ar3: (A4397)
KAS-FFC (TLSv1.2/v1.3)	KAS-Full	Full KAS-FFC key agreement used in TLSv1.2/v1.3 service	Bit-strength Caveat:Provides 112 or 128 bits encryption strength	KAS-FFC-SSC Sp800-56Ar3: (A4397) Domain Parameter Generation Method : ffdhe2048,

Name	Type	Description	Properties	Algorithms
				ffdhe3072 TLS v1.2 KDF RFC7627: (A4397) TLS v1.3 KDF: (A4397)
KAS-ECC (IKEv1/v2)	KAS-Full	Full KAS-ECC key agreement used in IKEv1/v2 service	Bit-strength Caveat:Provides between 128 and 256 bits encryption strength	KDF IKEv1: (A4397) KDF IKEv2: (A4397) KAS-ECC-SSC Sp800-56Ar3: (A4397)
KAS-FFC (IKEv1/v2)	KAS-Full	Full KAS-FFC key agreement used in IKEv1/v2 service	Bit-strength Caveat:Provides 112 or 128 bits of encryption strength	KDF IKEv1: (A4397) KDF IKEv2: (A4397) KAS-FFC-SSC Sp800-56Ar3: (A4397) Domain Parameter Generation Method: MODP-2048, MODP-3072
SSH RSA KeyGen	AsymKeyPair-KeyGen	RSA key gen		RSA KeyGen (FIPS186-5): (A4397) Counter DRBG: (A4397) CKG: ()
SSH RSA SigGen	DigSig-SigGen	RSA SigGen for SSHv2		RSA SigGen (FIPS186-5): (A4397)
SSH RSA SigVer	DigSig-SigVer	RSA SigVer for SSHv2		RSA SigVer (FIPS186-5): (A4397)
TLS ECDSA KeyGen	AsymKeyPair-KeyGen	ECDSA key gen for TLS		Counter DRBG: (A4397) ECDSA KeyGen (FIPS186-5): (A4397) CKG: ()
TLS ECDSA SigGen	DigSig-SigGen	ECDSA siggen for TLS		ECDSA SigGen (FIPS186-5): (A4397)

Name	Type	Description	Properties	Algorithms
TLS ECDSA SigVer	DigSig-SigVer	ECDSA sigver for TLS		ECDSA SigVer (FIPS186-5): (A4397)
TLS RSA KeyGen	AsymKeyPair-KeyGen	RSA key gen for TLS		Counter DRBG: (A4397) RSA KeyGen (FIPS186-5): (A4397) CKG: ()
TLS RSA SigGen	DigSig-SigGen	RSA siggen for TLS		RSA SigGen (FIPS186-5): (A4397)
TLS RSA SigVer	DigSig-SigVer	RSA sigver for TLS		RSA SigVer (FIPS186-5): (A4397)
IKEv1/v2 ECDSA KeyGen	AsymKeyPair-KeyGen	ECDSA keygen for IKEv1/v2		Counter DRBG: (A4397) ECDSA KeyGen (FIPS186-5): (A4397) CKG: ()
IKEv1/v2 ECDSA SigGen	DigSig-SigGen	ECDSA siggen for IKEv1/v2		ECDSA SigGen (FIPS186-5): (A4397)
IKEv1/v2 ECDSA SigVer	DigSig-SigVer	ECDSA SigVer		ECDSA SigVer (FIPS186-5): (A4397)
IKEv1/v2 RSA KeyGen	AsymKeyPair-KeyGen	RSA KeyGen for IKEv1/v2		Counter DRBG: (A4397) RSA KeyGen (FIPS186-5): (A4397) CKG: ()
IKEv1/v2 RSA SigGen	DigSig-SigGen	RSA SigGen for IKEv1/v2		RSA SigGen (FIPS186-5): (A4397)
IKEv1/v2 RSA SigVer	DigSig-SigVer	RSA SigVer for IKEv1/v2		RSA SigVer (FIPS186-5): (A4397)
SSH-KTS (AES-GCM)	KTS-Wrap	KTS wrap with AES-GCM	Bit-strength Caveat:Provides between 128 and 256 bits encryption strength	AES-GCM: (A4397)
SSH-KTS (AES and HMAC)	KTS-Wrap	KTS wrap with AES and HMAC	Bit-strength Caveat:Provides between 128 and 256 bits	AES-CTR: (A4397) HMAC-SHA-1: (A4397)

Name	Type	Description	Properties	Algorithms
			encryption strength	HMAC-SHA2-256: (A4397) HMAC-SHA2-384: (A4397) HMAC-SHA2-512: (A4397) SHA-1: (A4397) SHA2-256: (A4397) SHA2-384: (A4397) SHA2-512: (A4397)
TLS-KTS (AES-GCM)	KTS-Wrap	KTS wrap with AES-GCM	Bit-strength Caveat:Provides between 128 and 256 bits encryption strength	AES-GCM: (A4397)
TLS-KTS (AES and HMAC)	KTS-Wrap	TLS KTS wrap with AES and HMAC	Bit-strength Caveat:Provides between 128 and 256 bits encryption strength	AES-CBC: (A4397) HMAC-SHA2-256: (A4397) HMAC-SHA2-384: (A4397) HMAC-SHA2-512: (A4397) SHA2-256: (A4397) SHA2-384: (A4397) SHA2-512: (A4397)
Block ciphers (SSHv2)	BC-Auth BC-UnAuth MAC	Block ciphers used for SSHv2 service		AES-CTR: (A4397) HMAC-SHA-1: (A4397) HMAC-SHA2-256: (A4397) HMAC-SHA2-384: (A4397) HMAC-SHA2-512: (A4397) SHA-1: (A4397) SHA2-256: (A4397) SHA2-384: (A4397) SHA2-512:

Name	Type	Description	Properties	Algorithms
				(A4397) AES-GCM: (A4397)
Block ciphers (TLSv1.2/v1.3)	BC-Auth BC-UnAuth MAC	Block ciphers used for TLSv1.2/v1.3 service		AES-CBC: (A4397) AES-GCM: (A4397) HMAC-SHA2-256: (A4397) HMAC-SHA2-384: (A4397) HMAC-SHA2-512: (A4397) SHA2-256: (A4397) SHA2-384: (A4397) SHA2-512: (A4397)
Block ciphers (IKEv1/v2)	BC-Auth BC-UnAuth MAC	Block ciphers used in IKEv1/v2 serivce		AES-CBC: (A4397) AES-GCM: (A4397) HMAC-SHA-1: (A4397) HMAC-SHA2-256: (A4397) HMAC-SHA2-384: (A4397) HMAC-SHA2-512: (A4397) SHA-1: (A4397) SHA2-256: (A4397) SHA2-384: (A4397) SHA2-512: (A4397)
Firmware load test	MAC	Firmware load test		HMAC-SHA-1: (A4397) SHA-1: (A4397)
DRBG Function	DRBG	Used for DRBG generation		Counter DRBG: (A4397)

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

No specific algorithm information.

2.8 RBG and Entropy

Cert Number	Vendor Name
E160	WatchGuard Technologies, Inc.

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Firebox Cryptographic Jitter Entropy Module	Non-Physical	WatchGuard Firebox OS on Intel 6230, and WatchGuard Firebox OS on Intel E-2176G	256 bits	Full entropy	A4401 (SHA3-256)

Table 9: Entropy Sources

The module employs a Deterministic Random Bit Generator (DRBG) implementation based on SP800-90Arev1. This DRBG is used internally by the module (e.g. to generate symmetric keys, seeds for asymmetric key pairs, and random numbers for security functions).

The DRBG implemented is an AES-256 Counter DRBG, seeded by the entropy source described in the table above. The Counter DRBG utilizes the Derivation Function and implements prediction resistance feature.

The module's entropy source falls into IG 9.3.A, Scenario #1a: A hardware module with an entropy generating source inside the module's cryptographic boundary. Per the information from Public Use Document under ESV Cert. #E160, the validated module entropy source provides a full entropy output.

2.9 Key Generation

The module generates RSA, ECDSA, ECDH, and DH asymmetric key pairs compliant with FIPS 186-5, using a NIST SP 800-90Arev1 CTR DRBG for random number generation. In accordance with FIPS 140-3 IG D.H, the cryptographic module performs CKG for asymmetric keys as per section 5.1 of NIST SP 800-133rev2 (vendor affirmed) by obtaining a random bit string directly from an approved DRBG. The random bit string supports the required security strength requested by the calling application (without any V, as described in Additional Comments 2 of IG D.H.).

2.10 Key Establishment

The module provides the following key/SSP establishment services in the approved mode of operation:

- KAS-FFC Shared Secret Computation:

The module provides SP800-56Arev3 compliant key establishment according to FIPS 140-3 IG D.F scenario 2 path (2) with KAS-FFC shared secret computation. The shared secret computation provides 112 and or 128 bits of encryption strength. The module supports the use of the safe primes defined in RFC 4419 (SSH), RFC 7919 (TLS) and RFC 3526 (IKE).

  - SSH (RFC 4419):
    - MODP-2048 (ID = 14)
    - MODP-3072 (ID = 15)
  - TLS (RFC 7919):
    - ffdhe2048 (ID = 256)
    - ffdhe3072 (ID = 257)
  - IKE (RFC 3526):
    - MODP-2048 (ID = 14)
    - MODP-3072 (ID = 15)

- KAS-ECC Shared Secret Computation:
  The module provides SP800-56Arev3 compliant key establishment according to FIPS 140-3 IG D.F scenario 2 path (2) with KAS-ECC shared secret computation. The shared secret computation provides between 128 and 256 bits of encryption strength.

2.11 Industry Protocols

The module supports SSHv2, TLSv1.2, TLSv1.3, IKEv1 and IKEv2 industrial protocols. No parts of IKEv1, IKEv2, SSH and TLS protocols, other than the KDFs, have been tested by the CAVP and CMVP. Please refer to SSPs Table for more information.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
RJ-45 Management Interface, RJ-45 Ethernet Interfaces, and SFP+ Ethernet Interfaces	Data Input	Data input into the module for the services defined in Table 16 (Approved Services)
RJ-45 Management Interface, RJ-45 Ethernet Interfaces, and SFP+ Ethernet Interfaces	Data Output	Data output from the module for the services defined in Table 16 (Approved Services)
RJ-45 Management Interface, RJ-45 Ethernet Interfaces, SFP+ Ethernet Interfaces, and Reset Button	Control Input	Control input into the module for the services defined in Table 16 (Approved Services)
RJ-45 Management Interface, RJ-45 Ethernet Interfaces, SFP+ Ethernet Interfaces, Console Interface and LEDs	Status Output	Status information output from the module for the services defined in Table 16 (Approved Services)
Power Interface	Power	Power supply

Table 10: Ports and Interfaces

The module's physical perimeter encompasses the case of the tested platform mentioned in Table 2. The module provides physical ports which are mapped to logical interfaces provided by the module (data input, data output, control input, control output and status output) as above.

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Password-based authentication	The minimum length is eight (8) characters (94 possible characters). The probability that a random attempt will succeed or a false acceptance will occur is 1/(94^8) which is less than 1/1,000,000. As the module supports at most ten failed attempts to authenticate in a one-minute period, the probability of successfully authenticating to the module within one minute is 10/(94^8), which is less than 1/100,000. This calculation is based on the assumption that the typical standard American QWERTY computer keyboard has 10 Integer digits, 52 alphabetic characters, and 32 special characters providing 94 characters to choose from in total.	Password Based	The probability that a random attempt will succeed or a false acceptance will occur is 1/(94^8). Please refer to Description section in this table for more details	The probability of successfully authenticating to the module within one minute is 10/(94^8). Please refer to Description section in this table for more details

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
RSA-based authentication	The modules supports RSA public-key based authentication mechanism using a minimum of RSA 2048 bits, which provides 112 bits of security strength. The probability that a random attempt will succeed is 1/(2^112) which is less than 1/1,000,000. For multiple attacks during a one-minute period, as the module at its highest can support at most 17,000 new sessions per second to authenticate in a one-minute period, the probability of successfully authenticating to the module within a one minute period is 17,000 * 60 = 1,020,000/(2^112), which is less than 1/100,000.	RSA SigVer (FIPS186-5) (A4397)	The probability that a random attempt will succeed is 1/(2^112). Please refer to Description section in this table for more details	The probability of successfully authenticating to the module within a one minute period is 17,000 * 60 = 1,020,000/(2^112). Please refer to Description section in this table for more details
ECDSA-based authentication	The modules support ECDSA public-key based authentication mechanism using a minimum of curve P-256, which provides 128 bits of security strength. The probability that a random attempt will succeed is 1/(2^128) which is less than 1/1,000,000. For multiple attacks during a one-minute period, as the module at its highest can support at most 17,000 new	ECDSA SigVer (FIPS186-5) (A4397)	The probability that a random attempt will succeed is 1/(2^128) which is less than 1/1,000,000. Please refer to Description section in this table for more details	The probability of successfully authenticating to the module within a one minute period is 17,000 * 60 = 1,020,000/(2^128). Please refer to Description section in this table for more details

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	sessions per second to authenticate in a one-minute period, the probability of successfully authenticating to the module within a one minute period is 17,000 * 60 = 1,020,000/(2^128), which is less than 1/100,000.			

Table 11: Authentication Methods

The module implements identity-based authentication. The module supports Crypto Officer role User role and VPN Peer role. The module also allows the concurrent operators.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Identity	Crypto Officer	Password-based authentication
User	Identity	User	Password-based authentication
VPN Peer	Identity	User	RSA-based authentication ECDSA-based authentication

Table 12: Roles

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Roles account management	Create and manage roles accounts	N/A	Commands to create roles accounts	Status of the completion of account status	None	Crypto Officer - Crypto Officer password: G,W,Z - User passwo

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						rd: G,W,Z
Crypto Officer authentication	Crypto officer authentication to the module	N/A	Crypto Officer authentication request	Status of Cyrpto Officer authentication	None	Crypto Officer - Crypto Officer password: W,E
User authentication	User authentication to the module	N/A	User authentication request	Status of User authentication	None	User - User password: W,E
VPN Peer authentication	VPN Peer authentication	N/A	VPN Peer authentication request	Status of the VPN Peer authentication	IKEv1/v2 ECDSA SigVer IKEv1/v2 RSA SigVer	VPN Peer - IPSec/IKE RSA public key: W,E - IPSec/IKE ECDSA public key: W,E
Configure network	Configure module's network	N/A	Commands to configure the network	Status of the completion of network configuration status	None	Crypto Officer
Show status	Show module's status	N/A	Command used to show Module's Status	Module's operational status	None	Crypto Officer User
Show version	Show module's ID and versioning information	N/A	Command to show Module's ID and version	Module's ID and versioning information	None	Crypto Officer User

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Enable/Disable module's approved mode	Enable/disable module's approved mode serivce	N/A	Commands to enable/disable module's approved mode	Status of the module's mode status	None	Crypto Officer
Reboot module	Reboot module serivce	N/A	Commands to reboot the module	Status of the module's reboot status	None	Crypto Officer
Perform self-test	Perform self-tests serivce	N/A	Command to trigger self-tests	Status of the self-tests results	None	Crypto Officer
Firmware update	Perform firmware update serivce	Log showing approved algorithm used in firmware update service	Command to trigger firmware update	Status of the updated firmware installation	Firmware load test	Crypto Officer - Firmware load test key: R,E
Configure Bypass capability	Sets the Bypass capability	N/A	Commands to configure the Bypass capability	Status of the completion of Bypass capability configuration	None	Crypto Officer
Configure SSHv2 Function	Configure SSHv2 service	Logs showing approved algorithms used in SSHv2 service	Commands to configure SSHv2 service	Status of the completion of SSHv2 configuration	KAS-ECC-KeyGen (SSHv2) KAS-FFC-KeyGen (SSHv2) KAS-ECC (SSHv2) KAS-FFC (SSHv2) SSH RSA KeyGen SSH RSA SigGen SSH RSA SigVer Block ciphers (SSHv2)	Crypto Officer - DRBG entropy input: G,R,W,E - DRBG seed: G,R,W,E - DRBG internal state V value: G,R,W,

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					DRBG Function	E - DRBG key: G,R,W, E - SSH ECDH private key: G,R,W, E - SSH ECDH public key: G,R,W, E - SSH peer ECDH public key: G,R,W, E - SSH ECDH shared secret: G,R,W, E - SSH DH private key: G,R,W, E - SSH DH public key: G,R,W, E - SSH peer DH public key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,R,W,E - SSH DH shared secret: G,R,W,E - SSH RSA private key: G,R,W,E - SSH RSA public key: G,R,W,E - SSH encryption key: G,R,W,E - SSH integrity key: G,R,W,E User - DRBG entropy input: G,R,W,E - DRBG seed: G,R,W,E - DRBG internal state V value: G,R,W,

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - DRBG key: G,R,W, E - SSH ECDH private key: G,R,W, E - SSH ECDH public key: G,R,W, E - SSH peer ECDH public key: G,R,W, E - SSH ECDH shared secret: G,R,W, E - SSH DH private key: G,R,W, E - SSH DH public key: G,R,W, E - SSH peer DH public key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,R,W,E - SSH DH shared secret: G,R,W,E - SSH RSA private key: G,R,W,E - SSH RSA public key: G,R,W,E - SSH encryption key: G,R,W,E - SSH integrity key: G,R,W,E
Configure TLS (v1.2/v1.3) Function	Configure TLS (v1.2/v1.3) Function	Logs showing approved algorithms in TLS (v1.2/v1.3) service	Commands to configure TLS (v1.2/v1.3) service	Status of the completion of TLS (v1.2/v1.3) configuration	KAS-ECC-KeyGen (TLSv1.2/v1.3) KAS-FFC-KeyGen (TLSv1.2/v1.3) KAS-ECC (TLSv1.2/v1.3) KAS-FFC (TLSv1.2/v1.3) TLS ECDSA KeyGen TLS	Crypto Officer - DRBG entropy input: G,R,W,E - DRBG seed: G,R,W,E - DRBG internal state V value:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					ECDSA SigGen TLS ECDSA SigVer TLS RSA KeyGen TLS RSA SigGen TLS RSA SigVer Block ciphers (TLSv1.2/v1.3) DRBG Function	G,R,W,E - DRBG key: G,R,W,E - TLS ECDH private key: G,R,W,E - TLS ECDH public key: G,R,W,E - TLS peer ECDH public key: G,R,W,E - TLS ECDH shared secret: G,R,W,E - TLS DH private key: G,R,W,E - TLS DH public key: G,R,W,E - TLS peer DH public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key: G,R,W,E - TLS DH shared secret: G,R,W,E - TLS ECDSA private key: G,R,W,E - TLS ECDSA public key: G,R,W,E - TLS RSA private key: G,R,W,E - TLS RSA public key: G,R,W,E - TLS master secret: G,R,W,E - TLS encryption key: G,R,W,E - TLS integrity key: G,R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure IPsec/IKE (v1/v2)function	Configure IPsec/IKE (v1/v2)function	Logs showing approved algorithms used in IPsec/IKE (v1/v2) service	Commands to configure IPsec/IKE (v1/v2)function	Status of the completion of IPsec/IKE (v1/v2)configuration	KAS-ECC-KeyGen (IKEv1/v2) KAS-FFC-KeyGen (IKEv1/v2) KAS-ECC (IKEv1/v2) KAS-FFC (IKEv1/v2) IKEv1/v2 ECDSA KeyGen IKEv1/v2 ECDSA SigGen IKEv1/v2 ECDSA SigVer IKEv1/v2 RSA KeyGen IKEv1/v2 RSA SigGen IKEv1/v2 RSA SigVer Block ciphers (IKEv1/v2) DRBG Function	Crypto Officer - DRBG entropy input: G,R,W,E - DRBG seed: G,R,W,E - DRBG internal state V value: G,R,W,E - DRBG key: G,R,W,E - IPSec/IKE ECDH private key: G,R,W,E - IPSec/IKE peer ECDH public key: G,R,W,E - IPSec/IKE ECDH shared secret:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,R,W,E - IPsec/IKE DH private key: G,R,W,E - IPSec/IKE DH public key: G,R,W,E - IPSec/IKE peer DH public key: G,R,W,E - IPSec/IKE DH shared secret: G,R,W,E - IPSec/IKE ECDSA private key: G,R,W,E - IPSec/IKE ECDSA public key: G,R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- IPSec/IKE RSA private key: G,R,W,E - IPSec/IKE RSA public key: G,R,W,E - IPSec/IKE pre-shared secret: G,R,W,E - IPSec/IKE encryption key: G,R,W,E - IPSec/IKE integrity key: G,R,W,E
Perform Zeroization	Zeroize all SSPs	N/A	Command to zeroize the module	Status of the SSPs zeroization	None	Crypto Officer - DRBG entropy input: Z - DRBG seed: Z - DRBG

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						internal state V value: Z - DRBG key: Z - Crypto Officer password: Z - User password: Z - Firmware load test key: Z - SSH ECDH private key: Z - SSH ECDH public key: Z - SSH peer ECDH public key: Z - SSH ECDH shared secret: Z - SSH DH private key: Z - SSH DH public key: Z - SSH peer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						DH public key: Z - SSH DH shared secret: Z - SSH RSA private key: Z - SSH RSA public key: Z - SSH encryption key: Z - SSH integrity key: Z - TLS ECDH private key: Z - TLS ECDH public key: Z - TLS peer ECDH public key: Z - TLS ECDH shared secret: Z - TLS DH private key: Z - TLS DH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						public key: Z - TLS peer DH public key: Z - TLS DH shared secret: Z - TLS ECDSA private key: Z - TLS ECDSA public key: Z - TLS RSA private key: Z - TLS RSA public key: Z - TLS master secret: Z - TLS encryption key: Z - TLS integrity key: Z - IPSec/IKE ECDH private key: Z - IPSec/I

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						KE ECDH public key: Z - IPSec/IKE peer ECDH public key: Z - IPSec/IKE ECDH shared secret: Z - IPsec/IKE DH private key: Z - IPSec/IKE DH public key: Z - IPSec/IKE peer DH public key: Z - IPSec/IKE DH shared secret: Z - IPSec/IKE ECDSA private key: Z -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						IPSec/IKE ECDSA public key: Z - IPSec/IKE RSA private key: Z - IPSec/IKE RSA public key: Z - IPSec/IKE pre-shared secret: Z - IPSec/IKE encryption key: Z - IPSec/IKE integrity key: Z
Run SSHv2 Function	Run SSHv2 function	Logs showing approved algorithms used in SSHv2 service	Initiate SSHv2 session establishment request	Status of SSHv2 session establishment	KAS-ECC-KeyGen (SSHv2) KAS-FFC-KeyGen (SSHv2) KAS-ECC (SSHv2) KAS-FFC (SSHv2) SSH RSA KeyGen SSH RSA	Crypto Officer - DRBG entropy input: G,R,W,E - DRBG seed: G,R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					SigGen SSH RSA SigVer SSH-KTS (AES-GCM) SSH-KTS (AES and HMAC) Block ciphers (SSHv2) DRBG Function	- DRBG internal state V value: G,R,W, E - DRBG key: G,R,W, E - SSH ECDH private key: G,R,W, E - SSH ECDH public key: G,R,W, E - SSH peer ECDH public key: G,R,W, E - SSH ECDH shared secret: G,R,W, E - SSH DH private key: G,R,W, E - SSH DH public key: G,R,W,

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - SSH peer DH public key: G,R,W, E - SSH DH shared secret: G,R,W, E - SSH RSA private key: G,R,W, E - SSH RSA public key: G,R,W, E - SSH encryption key: G,R,W, E - SSH integrity key: G,R,W, E User - DRBG entropy input: G,R,W, E - DRBG seed: G,R,W, E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- DRBG internal state V value: G,R,W, E - DRBG key: G,R,W, E - SSH ECDH private key: G,R,W, E - SSH ECDH public key: G,R,W, E - SSH peer ECDH public key: G,R,W, E - SSH ECDH shared secret: G,R,W, E - SSH DH private key: G,R,W, E - SSH DH public key: G,R,W,

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - SSH peer DH public key: G,R,W, E - SSH DH shared secret: G,R,W, E - SSH RSA private key: G,R,W, E - SSH RSA public key: G,R,W, E - SSH encryption key: G,R,W, E - SSH integrity key: G,R,W, E
Run TLS (v1.2/v1.3) Function	Run TLS (v1.2/v1.3) function	Logs showing approved algorithms used in TLS (v1.2/v1.3)service	Initiate TLS (v1.2/v1.3) session establishment request	Status of TLS (v1.2/v1.3) session establishment	KAS-FFC-KeyGen (TLSv1.2/v1.3) KAS-ECC (TLSv1.2/v1.3) KAS-FFC (TLSv1.2/v1.3) TLS ECDSA	Crypto Officer - DRBG entropy input: G,R,W, E - DRBG seed: G,R,W,

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					KeyGen TLS ECDSA SigGen TLS ECDSA SigVer TLS RSA KeyGen TLS RSA SigGen TLS RSA SigVer TLS-KTS (AES-GCM) TLS-KTS (AES and HMAC) Block ciphers (TLSv1.2/v1.3) DRBG Function	E - DRBG internal state V value: G,R,W,E - DRBG key: G,R,W,E - TLS ECDH private key: G,R,W,E - TLS ECDH public key: G,R,W,E - TLS peer ECDH public key: G,R,W,E - TLS ECDH shared secret: G,R,W,E - TLS DH private key: G,R,W,E - TLS DH public key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,R,W,E - TLS peer DH public key: G,R,W,E - TLS DH shared secret: G,R,W,E - TLS ECDSA private key: G,R,W,E - TLS ECDSA public key: G,R,W,E - TLS RSA private key: G,R,W,E - TLS RSA public key: G,R,W,E - TLS master secret: G,R,W,E - TLS encryption key: G,R,W,

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - TLS integrity key: G,R,W, E User - DRBG entropy input: G,R,W, E - DRBG seed: G,R,W, E - DRBG internal state V value: G,R,W, E - DRBG key: G,R,W, E - TLS ECDH private key: G,R,W, E - TLS ECDH public key: G,R,W, E - TLS peer ECDH public key: G,R,W,

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - TLS ECDH shared secret: G,R,W, E - TLS DH private key: G,R,W, E - TLS DH public key: G,R,W, E - TLS peer DH public key: G,R,W, E - TLS DH shared secret: G,R,W, E - TLS ECDSA private key: G,R,W, E - TLS ECDSA public key: G,R,W, E - TLS RSA private key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,R,W,E - TLS RSA public key: G,R,W,E - TLS master secret: G,R,W,E - TLS encryption key: G,R,W,E - TLS integrity key: G,R,W,E
Run IPsec/IKE (v1/v2) Function	Run IPsec/IKE (v1/v2) function	Logs showing approved algorithms used in IPsec/IKE (v1/v2)service	Initiate IPsec/IKE (v1/v2) session establishment request	Status of IPSec/IKE (v1/v2) session establishment	KAS-ECC-KeyGen (IKEv1/v2) KAS-FFC-KeyGen (IKEv1/v2) KAS-ECC (IKEv1/v2) KAS-FFC (IKEv1/v2) IKEv1/v2 ECDSA KeyGen IKEv1/v2 ECDSA SigGen IKEv1/v2 ECDSA SigVer IKEv1/v2 RSA KeyGen IKEv1/v2 RSA SigGen	Crypto Officer - DRBG entropy input: G,R,W,E - DRBG seed: G,R,W,E - DRBG internal state V value: G,R,W,E - DRBG key: G,R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					IKEv1/v2 RSA SigVer Block ciphers (IKEv1/v2) DRBG Function	- IPSec/IKE ECDH private key: G,R,W,E - IPSec/IKE peer ECDH public key: G,R,W,E - IPSec/IKE ECDH shared secret: G,R,W,E - IPsec/IKE DH private key: G,R,W,E - IPSec/IKE DH public key: G,R,W,E - IPSec/IKE peer DH public key: G,R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- IPSec/IKE DH shared secret: G,R,W,E - IPSec/IKE ECDSA private key: G,R,W,E - IPSec/IKE ECDSA public key: G,R,W,E - IPSec/IKE RSA private key: G,R,W,E - IPSec/IKE RSA public key: G,R,W,E - IPSec/IKE pre-shared secret: G,R,W,E - IPSec/I

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						KE encryption key: G,R,W,E - IPSec/IKE integrity key: G,R,W,E User - DRBG entropy input: G,R,W,E - DRBG seed: G,R,W,E - DRBG internal state V value: G,R,W,E - DRBG key: G,R,W,E - IPSec/IKE ECDH private key: G,R,W,E - IPSec/IKE peer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						ECDH public key: G,R,W,E - IPSec/IKE ECDH shared secret: G,R,W,E - IPsec/IKE DH private key: G,R,W,E - IPSec/IKE DH public key: G,R,W,E - IPSec/IKE peer DH public key: G,R,W,E - IPSec/IKE DH shared secret: G,R,W,E - IPSec/IKE ECDSA private

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key: G,R,W,E - IPSec/IKE ECDSA public key: G,R,W,E - IPSec/IKE RSA private key: G,R,W,E - IPSec/IKE RSA public key: G,R,W,E - IPSec/IKE pre-shared secret: G,R,W,E - IPSec/IKE encryption key: G,R,W,E - IPSec/IKE integrity key: G,R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						VPN Peer - DRBG entropy input: G,R,W, E - DRBG seed: G,R,W, E - DRBG internal state V value: G,R,W, E - DRBG key: G,R,W, E - IPSec/IKE ECDH private key: G,R,W, E - IPSec/IKE peer ECDH public key: G,R,W, E - IPSec/IKE ECDH shared secret:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,R,W,E - IPsec/IKE DH private key: G,R,W,E - IPSec/IKE DH public key: G,R,W,E - IPSec/IKE peer DH public key: G,R,W,E - IPSec/IKE DH shared secret: G,R,W,E - IPSec/IKE ECDSA private key: G,R,W,E - IPSec/IKE ECDSA public key: G,R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- IPSec/IKE RSA private key: G,R,W,E - IPSec/IKE RSA public key: G,R,W,E - IPSec/IKE pre-shared secret: G,R,W,E - IPSec/IKE encryption key: G,R,W,E - IPSec/IKE integrity key: G,R,W,E
Run Bypass Function	Execute Bypass capability	N/A	Command to execute Bypass capability	Status of Bypass capability	None	Crypto Officer User

Table 13: Approved Services

4.4 Non-Approved Services

Name	Description	Algorithms	Role
MD5	Message digest in IPSec/IKE implementation in Non-approved mode	MD5	CO
DES	Data encryption/decryption in IPSec/IKE implementation in Non-approved mode	DES	CO
Triple-DES	Data encryption/decryption in IPSec/IKE implementation in Non-approved mode	Triple-DES	CO

Table 14: Non-Approved Services

4.5 External Software/Firmware Loaded

The module also supports the firmware update service. The version signature is verified by HMAC-SHA-1 (HMAC Cert. #A4397). A Firmware Load Test Key was preloaded to the module's binary at the factory and used for firmware load test. In order to load new firmware, the Crypto Officer must authenticate to the module before loading the firmware. This ensures that unauthorized access and use of the module is not performed. The module will load the new update upon reboot. The update attempt will be rejected if the verification fails.

4.6 Bypass Actions and Status

The module implements alternating Bypass service. Traffic output from the module's data output interface can be cryptographically protected via secure connections (SSH, TLS, or IPSec-VPN), or passed in plaintext (Bypass state), depending on Module's Bypass configuration. If the destined IP address is not covered by the module's configuration, the module would drop the traffic.

In addition, before running the Bypass service, the module would conduct two independent internal actions to activate the capability to prevent the inadvertent bypass of plaintext data due to a single error. If Bypass test succeeds, the module would proceed sending out the traffic in plaintext. If fails, the module would enter the error state, and drop the traffic.

4.7 Cryptographic Output Actions and Status

The module implements Self-initiated cryptographic output capability without external operator request. The Crypto Officer shall configure self-initiated cryptographic output capability. Prior to executing the self-initiated cryptographic output capability, the module conducts two independent internal actions to activate the capability to prevent the inadvertent output due to a single error.

4.8 Additional Information

The module supports Unauthenticated service, where the unauthenticated users can run the self-test service by power-cycling the module.

5 Software/Firmware Security

5.1 Integrity Techniques

The module is provided in the form of binary executable code. To ensure firmware security, the module is protected by ECDSA using Curve P-256 with SHA2-256 (ECDSA Cert. #A4397) algorithm. A Firmware Integrity Test Key (non-SSP) was preloaded to the module's binary at the factory and used for firmware integrity test only at the pre-operational self-test. The module uses the ECDSA with P-256 public key to verify the digital signature. If the firmware integrity test fails, the module would enter to an Error state with all crypto functionality inhibited.

5.2 Initiate on Demand

Integrity test is performed as part of the Pre-Operational Self-Tests. It is automatically executed at power-on. The operator can power-cycle or reboot the tested platform to initiate the firmware integrity test on-demand.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Tamper Evidence Labels (Part number: SKU WG8566)	90 days	Tamper evidence labels should be checked for nicks and scratches that make the metal case visible through the nicked or scratched seal. Tamper Evidence Label (TEL) may show any of the following as evidence of tampering or removal: TEL is not preset in the positions prescribed (as shown above); TEL has been cut; TEL is not stuck down well, or is loose; Self-destruction of the TEL (broken bits or shreds) present as from an attempt of removal
Production grade components	N/A	N/A

Table 15: Mechanisms and Actions Required

The following section demonstrates how to apply the tamper evident labels (TELs) to each module. The enclosure of the modules is the same.

The tamper evident labels shall be installed on the security devices containing the module prior to operating in the Approved mode. TELs shall be applied as depicted in the figures below. Any

unused TELs must be securely stored, accounted for, and maintained by the CO in a protected location.

Should the CO have to remove, change or replace TELs (tamper-evidence labels) for any reason, the CO must examine the location from which the TEL was removed and ensure that no residual debris is still remaining on the chassis or card. If residual debris remains, the CO must remove the debris using a damp cloth.

Any deviation of the TELs placement by unauthorized operators such as tearing, misconfiguration, removal, change, replacement or any other change in the TELs from its original configuration as depicted below shall mean the module is no longer in the Approved mode of operation. Returning the system back to the Approved mode of operation requires the replacement of the TELs as depicted below and any additional requirement per the site security policy which are out of scope of this Security Policy.

Tamper evident labels shall be applied for the module to operate in an approved mode of operation. It is the responsibility of the Cryptographic Office to properly place all tamper evident labels as described in this section, and the Cryptographic Officer should maintain control of unused labels in a secure location. The security labels recommended for FIPS 140-3 compliance are separately ordered (SKU WG8566). These security labels are designed to be very fragile and cannot be removed without visible signs of damage to the labels.

The Cryptographic Officer must apply tamper evident labels at the locations shown in the Figures below. Before the labels are applied, the Cryptographic Officer should ensure that the surface is clean. The surface should be cleaned using isopropyl alcohol and dried before applying the labels. After the labels are placed, the Cryptographic Officer should inspect the tamper evident labels periodically to verify they are intact.

If the tamper evident seals are found to be damaged or broken during inspection, the Cryptographic Officer can return the cryptographic module to an approved mode of operation by restoring the module to a factory default state, reinstalling, and applying new tamper evident labels.

Any attempt to open the device will damage the tamper evident seals or the material of the security appliance cover. Tamper evident seals can also be inspected for signs of tampering, which include the following: curled corners, rips, and slices.

The following is an example of WatchGuard Firebox Tamper Evident Labels that are used.

Figure 5: WatchGuard Firebox TEL sample

7.2 User Placed Tamper Seals

TEL quantity required for each module (Part number: SKU WG8566)

- M4800: 4 TELs
- M5800: 8 TELs

Placement:

Firebox M4800

Four (40 TELS are required.

Figure 6: M4800 Front View

Figure 7: M4800 Right View

Figure 8: M4800 Back View

Firebox M5800

Eight (8) TELs are required.

Figure 9: M5800 Front View

Figure 10: M5800 Left View

Figure 11: M5800 Right View

Figure 12: M5800 Back View

Surface Preparation: Clean the chassis of any grease, dirt, or oil before applying the tamper evident labels. Alcohol-based cleaning pads are recommended for this purpose.

Operator Responsible for Securing Unused Seals: Must be stored in a secure location under controlled access

8 Non-Invasive Security

N/A for this module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
DRAM	Volatile memory	Dynamic
Flash	Non-Volatile memory	Static

Table 16: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Module public key output	Module	External (Outside the Module's Boundary)	Plaintext	Automated	Electronic	
Peer public key input	External (Outside the Module's Boundary)	Module	Plaintext	Automated	Electronic	
SSPs Input/Output protected by TLS-KTS (AES-GCM)	External (Outside the Module's Boundary)	Module	Encrypted	Automated	Electronic	TLS-KTS (AES-GCM)
SSPs Input/Output protected by TLS-KTS (AES and HMAC)	External (Outside the Module's Boundary)	Module	Encrypted	Automated	Electronic	TLS-KTS (AES and HMAC)
SSPs Input/Output protected by SSH-KTS (AES-GCM)	External (Outside the Module's Boundary)	Module	Encrypted	Automated	Electronic	SSH-KTS (AES-GCM)

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
SSPs Input/Output protected by SSH-KTS (AES and HMAC)	External (Outside the Module's Boundary)	Module	Encrypted	Automated	Electronic	SSH-KTS (AES and HMAC)

Table 17: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Zeroization command	CO issues zeroization service: "fips zeroize" to zeroize all SSPs	The zeroization command will erase all SSPs stored in Module's DRAM or Flash memory	CO issues zeroization command
Session termination	Zeroization upon session termination	Session termination will automatically zeroize all session based temporary SSPs	Terminate session
Reboot	Zeroization upon rebooting the module	Reboot to zeroize all temporary SSPs stored in Module's DRAM	Reboot

Table 18: SSP Zeroization Methods

Please note that the Firmware Load Test Key is only used for Firmware Load Test, and not subject to the zeroization requirement.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG entropy input	Used to seed the DRBG	384 bits - 384 bits	Entropy inputs - CSP			DRBG Function
DRBG seed	Used for DRBG generation	384 bits - 384 bits	DRBG parameters - CSP			DRBG Function
DRBG internal state V value	Used for DRBG generation	384 bits - 384 bits	DRBG parameters - CSP			DRBG Function
DRBG key	Used for DRBG generation	256 bits - 256 bits	DRBG parameter - CSP			DRBG Function

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Crypto Officer password	Used for Crypto Officer authentication	8-32 characters - N/A	Authentication Data - CSP			
User password	Used for User authentication	8-32 characters - N/A	Authentication Data - CSP			
Firmware load test key	Used for firmware load test	at least 112 bits - at least 112 bits	MAC Key - Neither			Firmware load test
SSH ECDH private key	Used for SSH shared secret derivation	Curves: P-256, P-384 and P-512 - 128-256 bits	Private Key - CSP	KAS-ECC-KeyGen (SSHv2)		KAS-ECC-KeyGen (SSHv2)
SSH ECDH public key	Used for SSH peer shared secret derivation	Curves: P-256, P-384 and P-512 - 128-256 bits	Public Key - PSP		KAS-ECC-KeyGen (SSHv2)	
SSH peer ECDH public key	Used for SSH shared secret derivation	Curves: P-256, P-384 and P-512 - N/A	Public Key - PSP			KAS-ECC-KeyGen (SSHv2)
SSH ECDH shared secret	Used for SSH encryption key and SSH integrity key derivation	Curves: P-256, P-384 and P-512 - 128-256 bits	Shared Secret - CSP		KAS-ECC (SSHv2)	KAS-ECC (SSHv2)
SSH DH private key	Used for SSH shared secret derivation	MODP-2048 and MODP-3072 - 112-128 bits	Private Key - CSP	KAS-FFC-KeyGen (SSHv2)		KAS-FFC-KeyGen (SSHv2)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SSH DH public key	Used for SSH peer shared secret derivation	MODP-2048 and MODP-3072 - 112-128 bits	Public Key - PSP		KAS-FFC-KeyGen (SSHv2)	
SSH peer DH public key	Used to derive SSH DH shared secret	MODP-2048 and MODP-3072 - N/A	Public Key - PSP			KAS-FFC-KeyGen (SSHv2)
SSH DH shared secret	Used for SSH encryption key and SSH integrity key derivation	MODP-2048 and MODP-3072 - 128-256 bits	Shared Secret - CSP		KAS-FFC (SSHv2)	KAS-FFC (SSHv2)
SSH RSA private key	Used for SSH authentication	Modulus: 2048, 3072 and 4096 bits - 112-152 bits	Private Key - CSP	SSH RSA KeyGen		SSH RSA SigGen
SSH RSA public key	Used for SSH authentication	Modulus: 2048, 3072 and 4096 bits - 112-152 bits	Public Key - PSP		SSH RSA KeyGen	SSH RSA SigVer
SSH encryption key	Used for SSH traffic confidentiality protection	128-256 bits - 128-256 bits	Symmetric Key - CSP		KAS-ECC (SSHv2) KAS-FFC (SSHv2)	Block ciphers (SSHv2)
SSH integrity key	Used to for SSH traffic integrity protection	at least 112 bits - at least 112 bits	Integrity Key - CSP		KAS-ECC (SSHv2) KAS-FFC (SSHv2)	Block ciphers (SSHv2)
TLS ECDH	Used for TLS shared	Curves: P-256,	Private Key - CSP	KAS-ECC-KeyGen		KAS-ECC-KeyGen

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
private key	secret derivation	P-384 and P-512 - 128-256 bits		(TLSv1.2/v1.3)		(TLSv1.2/v1.3)
TLS ECDH public key	Used for TLS peer shared secret derivation	Curves: P-256, P-384 and P-512 - 128-256 bits	Public Key - PSP		KAS-ECC-KeyGen (TLSv1.2/v1.3)	
TLS peer ECDH public key	Used for TLS ECDH shared secret derivation	Curves: P-256, P-384 and P-512 - N/A	Public Key - PSP			KAS-ECC-KeyGen (TLSv1.2/v1.3)
TLS ECDH shared secret	This CSP is also referred to TLS pre-master secret if EC Diffie-Hellman is used for TLS key agreement. This CSP is used for TLS master secret derivation	Curves: P-256, P-384 and P-512 - 128-256 bits	Shared Secret - CSP		KAS-ECC (TLSv1.2/v1.3)	KAS-ECC (TLSv1.2/v1.3)
TLS DH private key	Used for TLS DH shared secret derivation	ffdhe2048 and ffdhe3072 - 112-128 bits	Private Key - CSP	KAS-FFC-KeyGen (TLSv1.2/v1.3)		KAS-FFC-KeyGen (TLSv1.2/v1.3)
TLS DH public key	Used for TLS peer DH shared secret derivation	ffdhe2048 and ffdhe3072 - 112-128 bits	Public Key - PSP		KAS-FFC-KeyGen (TLSv1.2/v1.3)	
TLS peer DH	Used to derive TLS	ffdhe2048 and	Public Key - PSP			KAS-FFC-KeyGen

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
public key	DH shared secret	ffdhe3072 - N/A				(TLSv1.2/v1.3)
TLS DH shared secret	This CSP is also referred to TLS pre-master secret if Diffie-Hellman is used for TLS key agreement. This CSP is used for TLS master secret derivation	ffdhe2048 and ffdhe3072 - 112-128 bits	Shared Secret - CSP		KAS-FFC (TLSv1.2/v1.3)	KAS-FFC (TLSv1.2/v1.3)
TLS ECDSA private key	Used for TLS authentication	Curves: P-256, P-384 and P-512 - 128-256 bits	Private Key - CSP	TLS ECDSA KeyGen		TLS ECDSA SigGen
TLS ECDSA public key	Used for TLS authentication	Curves: P-256, P-384 and P-512 - 128-256 bits	Public Key - PSP		TLS ECDSA KeyGen	TLS ECDSA SigVer
TLS RSA private key	Used for TLS authentication	Modulus: 2048, 3072 and 4096 bits - 112 -152 bits	Private Key - CSP	TLS RSA KeyGen		TLS RSA SigGen
TLS RSA public key	Used for TLS authentication	Modulus: 2048, 3072 and 4096 bits -	Public Key - PSP		TLS RSA KeyGen	TLS RSA SigVer

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		112 -152 bits				
TLS master secret	Used for TLS encryption key and TLS integrity key derivation	384 bits - N/A	TLS master secret - CSP		KAS-ECC (TLSv1.2/v1.3) KAS-FFC (TLSv1.2/v1.3)	KAS-ECC (TLSv1.2/v1.3) KAS-FFC (TLSv1.2/v1.3)
TLS encryption key	Used for TLS traffic confidentiality protection	128-256 bits - 128-256 bits	Symmetric Key - CSP		KAS-ECC (TLSv1.2/v1.3) KAS-FFC (TLSv1.2/v1.3)	Block ciphers (TLSv1.2/v1.3)
TLS integrity key	Used for TLS traffic integrity protection	at least 112 bits - at least 112 bits	Integrity Key - CSP		KAS-ECC (TLSv1.2/v1.3) KAS-FFC (TLSv1.2/v1.3)	Block ciphers (TLSv1.2/v1.3)
IPSec/IKE ECDH private key	Used for IPSec/IKE ECDH shared secret derivation	Curves: P-256, P-384 and P-521 - 128-256 bits	Private Key - CSP	KAS-ECC-KeyGen (IKEv1/v2)		KAS-ECC-KeyGen (IKEv1/v2)
IPSec/IKE ECDH public key	Used for IPSec/IKE peer ECDH shared secret derivation	Curves: P-256, P-384 and P-521 - 128-256 bits	Public Key - PSP		KAS-ECC-KeyGen (IKEv1/v2)	
IPSec/IKE peer ECDH public key	Used for IPSec/IKE ECDH shared secret derivation	Curves: P-256, P-384 and P-521 - N/A	Public Key - PSP			KAS-ECC-KeyGen (IKEv1/v2)
IPSec/IKE ECDH shared secret	Used for IPSec/IKE encryption key and IPSec/IKE	Curves: P-256, P-384 and P-521 -	Shared Secret - CSP		KAS-ECC (IKEv1/v2)	KAS-ECC (IKEv1/v2)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	integrity key derivation	128-256 bits				
IPsec/IKE DH private key	Used for IPSec/IKE DH shared secret derivation	MODP-2048 and MODP-3072 - 112-128 bits	Private Key - CSP	KAS-FFC-KeyGen (IKEv1/v2)		KAS-FFC-KeyGen (IKEv1/v2)
IPSec/IKE DH public key	Used for IPSec/IKE peer DH shared secret derivation	MODP-2048 and MODP-3072 - 112-128 bits	Public Key - PSP		KAS-FFC-KeyGen (IKEv1/v2)	
IPSec/IKE peer DH public key	Used to derive IKE DH shared secret	MODP-2048 and MODP-3072 - N/A	Public Key - PSP			KAS-FFC (IKEv1/v2)
IPSec/IKE DH shared secret	Used for IPSec/IKE encryption key and IPSec/IKE integrity key derivation	MODP-2048 and MODP-3072 - 112-128 bits	Shared Secret - CSP		KAS-FFC (IKEv1/v2)	KAS-FFC (IKEv1/v2)
IPSec/IKE ECDSA private key	Used for IPSec/IKE authentication	Curves: P-256, P-384 and P-521 - 128-256 bits	Private Key - CSP	IKEv1/v2 ECDSA KeyGen		IKEv1/v2 ECDSA SigGen
IPSec/IKE ECDSA public key	Used for IPSec/IKE authentication	Curves: P-256, P-384 and P-521 - 128-256 bits	Public Key - PSP		IKEv1/v2 ECDSA KeyGen	IKEv1/v2 ECDSA SigVer
IPSec/IKE RSA private key	Used for IPSec/IKE authentication	Modulus: 2048, 3072 and	Private Key - CSP	IKEv1/v2 RSA KeyGen		IKEv1/v2 RSA SigGen

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		4096 bits - 112-152 bits				
IPSec/IKE RSA public key	Used for IPSec/IKE authentication	Modulus: 2048, 3072 and 4096 bits - 112-152 bits	Public Key - PSP		IKEv1/v2 RSA KeyGen	IKEv1/v2 RSA SigVer
IPSec/IKE pre-shared secret	Used for IPSec/IKE peer authentication	16-32 bytes characters - N/A	Shared secret - CSP			
IPSec/IKE encryption key	Used for IPSec/IKE traffic confidentiality protection	128-256 bits - 128-256 bits	Symmetric key - CSP		KAS-ECC (IKEv1/v2) KAS-FFC (IKEv1/v2)	Block ciphers (IKEv1/v2)
IPSec/IKE integrity key	Used for IPSec/IKE traffic integrity protection	At least 112 bits - At least 112 bits	Integrity key - CSP		KAS-ECC (IKEv1/v2) KAS-FFC (IKEv1/v2)	Block ciphers (IKEv1/v2)

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DRBG entropy input		DRAM:Plaintext	Until reboot	Zeroization command Session termination Reboot	DRBG seed:Used With DRBG internal state V value:Used With DRBG key:Used With
DRBG seed		DRAM:Plaintext	Until reboot	Zeroization command Session termination Reboot	DRBG entropy input:Used With DRBG internal state V value:Used With DRBG key:Used With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DRBG internal state V value		DRAM:Plaintext	Until reboot	Zeroization command Session termination Reboot	DRBG entropy input:Used With DRBG seed:Used With DRBG key:Used With
DRBG key		DRAM:Plaintext	Until Reboot	Zeroization command Session termination Reboot	DRBG entropy input:Used With DRBG seed:Used With DRBG internal state V value:Used With
Crypto Officer password	SSPs Input/Output protected by TLS-KTS (AES-GCM) SSPs Input/Output protected by TLS-KTS (AES and HMAC) SSPs Input/Output protected by SSH-KTS (AES-GCM) SSPs Input/Output protected by SSH-KTS (AES and HMAC)	Flash:Plaintext	Until zeroized	Zeroization command	Crypto Officer password:Used With
User password	SSPs Input/Output protected by TLS-KTS (AES-GCM) SSPs Input/Output protected by TLS-KTS (AES and HMAC) SSPs Input/Output	Flash:Plaintext	Until zeroized	Zeroization command	User password:Used With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	protected by SSH-KTS (AES-GCM) SSPs Input/Output protected by SSH-KTS (AES and HMAC)				
Firmware load test key		Flash:Plaintext	Until zeroized	N/A	Firmware load test key:Used With
SSH ECDH private key		DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH ECDH public key:Paired With SSH peer DH public key:Used With SSH DH shared secret:Derived to
SSH ECDH public key	Module public key output	DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH ECDH private key:Paired With
SSH peer ECDH public key	Peer public key input	DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH ECDH private key:Used With SSH ECDH shared secret:Derived to
SSH ECDH shared secret		DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH encryption key:Derived From SSH integrity key:Derived From
SSH DH private key		DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH DH public key:Paired With SSH peer DH public key:Used With SSH DH shared secret:Derived to
SSH DH public key	Module public key output	DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH DH private key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SSH peer DH public key	Peer public key input	DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH DH private key:Used With SSH DH shared secret:Derived From
SSH DH shared secret		DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH encryption key:Derived to SSH integrity key:Derived to
SSH RSA private key		Flash:Plaintext	while SSH session is on	Zeroization command	SSH RSA public key:Paired With
SSH RSA public key	SSPs Input/Output protected by TLS-KTS (AES-GCM) SSPs Input/Output protected by TLS-KTS (AES and HMAC) SSPs Input/Output protected by SSH-KTS (AES-GCM) SSPs Input/Output protected by SSH-KTS (AES and HMAC)	Flash:Plaintext	while SSH session is on	Zeroization command	SSH RSA private key:Paired With
SSH encryption key		DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH ECDH shared secret:Derived From SSH DH shared secret:Derived From
SSH integrity key		DRAM:Plaintext	while SSH session is on	Zeroization command Session termination Reboot	SSH ECDH shared secret:Derived From SSH DH shared

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					secret:Derived From
TLS ECDH private key		DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS ECDH public key:Paired With TLS peer ECDH public key:Used With
TLS ECDH public key	Module public key output	DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS ECDH private key:Paired With
TLS peer ECDH public key	Peer public key input	DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS ECDH private key:Used With
TLS ECDH shared secret		DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS ECDH private key:Derived From TLS peer ECDH public key:Derived From
TLS DH private key		DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS DH public key:Paired With TLS peer DH public key:Used With
TLS DH public key	Module public key output	DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS DH private key:Paired With
TLS peer DH public key	Peer public key input	DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS DH private key:Used With
TLS DH shared secret		DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS DH private key:Derived From TLS peer DH public key:Derived From
TLS ECDSA private key		Flash:Plaintext	while TLS session is on	Zeroization command	TLS ECDSA public key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS ECDSA public key	SSPs Input/Output protected by TLS-KTS (AES-GCM) SSPs Input/Output protected by TLS-KTS (AES and HMAC) SSPs Input/Output protected by SSH-KTS (AES-GCM) SSPs Input/Output protected by SSH-KTS (AES and HMAC)	Flash:Plaintext	while TLS session is on	Zeroization command	TLS ECDSA private key:Paired With
TLS RSA private key		Flash:Plaintext	while TLS session is on	Zeroization command	TLS RSA public key:Paired With
TLS RSA public key	SSPs Input/Output protected by TLS-KTS (AES-GCM) SSPs Input/Output protected by TLS-KTS (AES and HMAC) SSPs Input/Output protected by SSH-KTS (AES-GCM) SSPs Input/Output protected by SSH-KTS (AES and HMAC)	Flash:Plaintext	while TLS session is on	Zeroization command	TLS RSA private key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS master secret		DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS ECDH shared secret:Derived From
TLS encryption key		DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS integrity key:Used With
TLS integrity key		DRAM:Plaintext	while TLS session is on	Zeroization command Session termination Reboot	TLS encryption key:Used With
IPSec/IKE ECDH private key		DRAM:Plaintext	while IPSec/IKE session is on	Zeroization command Session termination Reboot	IPSec/IKE peer ECDH public key:Paired With
IPSec/IKE ECDH public key	Module public key output	DRAM:Plaintext	while IPSec/IKE session is on	Zeroization command Session termination Reboot	IPSec/IKE ECDH private key:Paired With
IPSec/IKE peer ECDH public key	Peer public key input	DRAM:Plaintext	while IPSec/IKE session is on	Zeroization command Session termination Reboot	IPSec/IKE ECDH private key:Used With
IPSec/IKE ECDH shared secret		DRAM:Plaintext	while IPSec/IKE session is on	Zeroization command Session termination Reboot	SKEYSEED:Used With IPSec/IKE encryption key:Derived to IPSec/IKE integrity key:Derived to
IPsec/IKE DH private key		DRAM:Plaintext	while IPSec/IKE session is on	Zeroization command Session termination Reboot	IPSec/IKE DH public key:Paired With
IPSec/IKE DH public key	Module public key output	DRAM:Plaintext	while IPSec/IKE session is on	Zeroization command Session termination Reboot	IPsec/IKE DH private key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
IPSec/IKE peer DH public key	Peer public key input	DRAM:Plaintext	while IPSec/IKE session is on	Zeroization command Session termination Reboot	IPsec/IKE DH private key:Used With
IPSec/IKE DH shared secret		DRAM:Plaintext	while IPSec/IKE session is on	Zeroization command Session termination Reboot	SSH ECDH shared secret:Derive to IPSec/IKE DH shared secret:Derive to
IPSec/IKE ECDSA private key		Flash:Plaintext	while IPSec/IKE session is on	Zeroization command	IPSec/IKE ECDSA public key:Paired With
IPSec/IKE ECDSA public key	SSPs Input/Output protected by TLS-KTS (AES-GCM) SSPs Input/Output protected by TLS-KTS (AES and HMAC) SSPs Input/Output protected by SSH-KTS (AES-GCM) SSPs Input/Output protected by SSH-KTS (AES and HMAC)	Flash:Plaintext	while IPSec/IKE session is on	Zeroization command	IPSec/IKE ECDSA private key:Paired With
IPSec/IKE RSA private key		Flash:Plaintext	while IPSec/IKE session is on	Zeroization command	IPSec/IKE RSA public key:Paired With
IPSec/IKE RSA public key	SSPs Input/Output protected by TLS-KTS (AES-GCM) SSPs Input/Output	Flash:Plaintext	while IPSec/IKE session is on	Zeroization command	IPSec/IKE RSA private key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	protected by TLS-KTS (AES and HMAC) SSPs Input/Output protected by SSH-KTS (AES-GCM) SSPs Input/Output protected by SSH-KTS (AES and HMAC)				
IPSec/IKE pre-shared secret	SSPs Input/Output protected by TLS-KTS (AES-GCM) SSPs Input/Output protected by TLS-KTS (AES and HMAC) SSPs Input/Output protected by SSH-KTS (AES-GCM) SSPs Input/Output protected by SSH-KTS (AES and HMAC)	Flash:Encrypted	until zeroized	Zeroization command	
IPSec/IKE encryption key		DRAM:Plaintext	while IPSec/IKEv session is on	Zeroization command Session termination Reboot	IPSec/IKE DH shared secret:Derived From IPSec/IKE ECDH shared secret:Derived From
IPSec/IKE integrity key		DRAM:Plaintext	while IPSec/IKE	Zeroization command Session	IPSec/IKE DH shared secret:Derived

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			session is on	termination Reboot	From IPSec/IKE ECDH shared secret:Derived From

Table 20: SSP Table 2

9.5 Transitions

SHA-1: The module includes an implementation of SHA-1 for hashing and digital signature verification. This implementation will be non-Approved for all uses starting January 1, 2031.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
ECDSA SigVer (FIPS186-5) (A4397)	P-256 with SHA2-256	KAT	SW/FW Integrity	Status message "Integrity check succeeded" is output to the log	Module performs ECDSA SigVer KAT prior to firmware integrity test
Pre-Operational Bypass Test	N/A	N/A	Bypass	Module is in normal state	Module performs pre-operational bypass test at start up

Table 21: Pre-Operational Self-Tests

The module performs the following self-tests, including the pre-operational self-tests and Conditional self-tests. Prior to the module providing any data output via the data output interface, the module performs and passes the pre-operational self-tests. Following the successful pre-operational self-tests, the module executes the Conditional Cryptographic Algorithm Self-tests (CASTs). If anyone of the self-tests fails, the module transitions into an error state and outputs the error message via the module's status output interface. While the module is in the error state, all data through the data output interface and all cryptographic operations are disabled. The error state can only be cleared by reloading the module. All self-tests must be completed successfully before the module transitions to the operational state.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC Encrypt KAT (A4397)	128 bits	KAT	CAST	Module is in normal state	Encryption	Power up
AES-CBC Decrypt KAT (A4397)	128 bits	KAT	CAST	Module is in normal state	Decryption	Power up
AES-GCM Authenticated Encrypt KAT (A4397	128 bits	KAT	CAST	Module is in normal state	Authenticated Encryption	Power up
AES-GCM Authenticated Decrypt KAT (A4397)	128 bits	KAT	CAST	Module is in normal state	Authenticated Decryption	Power up
Counter DRBG Instantiate KAT (A4397)	AES-128	KAT	CAST	Module is in normal state	CTR_DRBG Instantiate	Power up
Counter DRBG Generate KAT (A4397)	AES-128	KAT	CAST	Module is in normal state	CTR_DRBG Generate	Power up
Counter DRBG Reseed KAT (A4397)	AES-128	KAT	CAST	Module is in normal state	CTR_DRBG Reseed	Power up
ECDSA SigGen (FIPS186-5) KAT (A4397)	P-256 with SHA2-256	KAT	CAST	Module is in normal state	N/A	Power up
ECDSA SigVer (FIPS186-5) KAT (A4397)	P-256 with SHA2-256	KAT	CAST	Module is in normal state	N/A	Power up
KAS-ECC-SSC Sp800-56Ar3 KAT (A4397)	P-256 with SHA2-256	KAT	CAST	Module is in normal state	KAS-ECC-SSC Primitive Z	Power up
KAS-FFC-SSC Sp800-56Ar3 KAT (A4397)	MODP-2048	KAT	CAST	Module is in normal state	KAS-FFC-SSC Primitive Z	Power up
HMAC-SHA-1 KAT (A4397)	SHA-1	KAT	CAST	Module is in normal state	N/A	Power up
HMAC-SHA2-256 KAT (A4397)	SHA2-256	KAT	CAST	Module is in normal state	N/A	Power up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-384 KAT (A4397)	SHA2-384	KAT	CAST	Module is in normal state	N/A	Power up
HMAC-SHA2-512 KAT (A4397)	SHA2-512	KAT	CAST	Module is in normal state	N/A	Power up
RSA SigGen (FIPS186-5) KAT (A4397)	2048 bits	KAT	CAST	Module is in normal state	N/A	Power up
RSA SigVer (FIPS186-5) KAT (A4397)	2048 bits	KAT	CAST	Module is in normal state	N/A	Power up
KDF IKEv1 KAT (A4397)	N/A	KAT	CAST	Module is in normal state	N/A	Power up
KDF IKEv2 KAT (A4397)	N/A	KAT	CAST	Module is in normal state	N/A	Power up
KDF SSH KAT (A4397)	N/A	KAT	CAST	Module is in normal state	N/A	Power up
TLS v1.2 KDF RFC7627 (A4397)	N/A	KAT	CAST	Module is in normal state	N/A	Power up
TLS v1.3 KDF KAT (A4397)	N/A	KAT	CAST	Module is in normal state	N/A	Power up
Entropy Source Start-up Health Test (RCT)	N/A	RCT	CAST	Module is in normal state	N/A	Power up
Entropy Source Start-up Health Test (APT)	N/A	APT	CAST	Module is in normal state	N/A	Power up
Entropy Source Continuous Health Test (RCT)	N/A	RCT	CAST	Module is in normal state	N/A	Power up
Entropy Source Continuous Health Test (APT)	N/A	APT	CAST	Module is in normal state	N/A	Power up
KAS-ECC-SSC Sp800-	P-256 with SHA2-256	PCT	PCT	Module is in normal state	N/A	New KAS ECC Keypair generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
56Ar3 PCT (A4397)						
KAS-FFC-SSC Sp800-56Ar3 PCT (A4397)	MODP-2048	PCT	PCT	Module is in normal state	N/A	New KAS FFC Keypair generation
ECDSA KeyGen (FIPS186-5) PCT (A4397)	P-256 with SHA2-256	PCT	PCT	Module is in normal state	ECDSA	New ECDSA Keypair generation
RSA KeyGen (FIPS186-5) PCT (A4397)	2048 bits	PCT	PCT	Module is in normal state	RSA	New RSA Keypair generation
Firmware Load Test	HMAC-SHA-1	KAT	SW/FW Load	Module is in normal state	N/A	while doing the firmware upload test
Conditional Bypass	N/A	N/A	Bypass	Module is in normal state	N/A	Performs conditional bypass test before first operational use of bypass service

Table 22: Conditional Self-Tests

The module performs on-demand self-tests initiated by the operator, by powering off and powering the module back on. The full suite of self-tests is then executed. The same procedure may be employed by the operator to perform periodic self-tests.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigVer (FIPS186-5) (A4397)	KAT	SW/FW Integrity	Recommend 60 days	Module reboot
Pre-Operational Bypass Test	N/A	Bypass	Recommend 60 days	Module reboot

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC Encrypt KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC Decrypt KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
AES-GCM Authenticated Encrypt KAT (A4397	KAT	CAST	Recommend 60 days	Module reboot
AES-GCM Authenticated Decrypt KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
Counter DRBG Instantiate KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
Counter DRBG Generate KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
Counter DRBG Reseed KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
ECDSA SigGen (FIPS186-5) KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
ECDSA SigVer (FIPS186-5) KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
KAS-ECC-SSC Sp800-56Ar3 KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
KAS-FFC-SSC Sp800-56Ar3 KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
HMAC-SHA-1 KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
HMAC-SHA2-256 KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
HMAC-SHA2-384 KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
HMAC-SHA2-512 KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
RSA SigGen (FIPS186-5) KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-5) KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
KDF IKEv1 KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
KDF IKEv2 KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
KDF SSH KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
TLS v1.2 KDF RFC7627 (A4397)	KAT	CAST	Recommend 60 days	Module reboot
TLS v1.3 KDF KAT (A4397)	KAT	CAST	Recommend 60 days	Module reboot
Entropy Source Start-up Health Test (RCT)	RCT	CAST	Recommend 60 days	Module Reboot
Entropy Source Start-up Health Test (APT)	APT	CAST	Recommend 60 days	Module Reboot
Entropy Source Continuous Health Test (RCT)	RCT	CAST	Recommend 60 days	Module Reboot
Entropy Source Continuous Health Test (APT)	APT	CAST	Recommend 60 days	Module Reboot
KAS-ECC-SSC Sp800-56Ar3 PCT (A4397)	PCT	PCT	N/A	New KAS ECC Keypair generation
KAS-FFC-SSC Sp800-56Ar3 PCT (A4397)	PCT	PCT	N/A	New KAS FFC Keypair generation
ECDSA KeyGen (FIPS186-5) PCT (A4397)	PCT	PCT	N/A	New ECDSA Keypair generation
RSA KeyGen (FIPS186-5) PCT (A4397)	PCT	PCT	N/A	New RSA Keypair generation
Firmware Load Test	KAT	SW/FW Load	N/A	N/A
Conditional Bypass	N/A	Bypass	N/A	N/A

Table 24: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error State	If self-test tests fail, the module enters the error state	Self-tests failure	Reboot the module	Error message is output via the Console interface

Table 25: Error States

If any of the above-mentioned self-tests fail, the module reports the error and enters the Error state. In the Error State, no cryptographic services are provided, and data output is prohibited. The only method to recover from the error state is to reboot the module and perform the self-tests, including the pre-operational firmware integrity test and the conditional CASTs. The module will only enter into the operational state after successfully passing the pre-operational firmware integrity test and the conditional CASTs.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module meets all the Level 2 requirements for FIPS 140-3. Follow the secure operations provided below to place the module in approved mode. Operating this module without maintaining the following settings will remove the module from the approved mode of operation. Any firmware/software loaded into this module that is not shown on the module certificate, is out of the scope of this validation and requires a separate FIPS 140-3 validation.

Secure Installation
The Crypto Officer must ensure that:

- The Firebox module is installed in a secure physical location.
- Physical access to the Firebox module is restricted to authorized personnel only.

Secure Operation
The cryptographic module is not configured to operate in approved mode by default. To operate in Approved mode, the crypto officer shall strictly follow the following procedures.

- Apply tamper evident labels as described in section “Physical Security” of this document
- Assume Crypto Officer (CO) role by using the default credentials (User name: admin; Password: readwrite) if this is the first time use, and create the operator passwords with a minimum of 8 characters. Please note that the default password will not be accepted by the module after the initial access, and the CO needs to authenticate to the module every time that the CO needs to make a change to the configuration.
- Issue the CLI command ‘fips enable’.
- When configuring any and all IPSec VPN tunnels, the Crypto Officer can either accept the default setting (using AES256, SHA2-256 and DH-group 14), or select any

algorithms from Table 16. Below is an example to show that Crypto Officer changed the configuration from default settings to use AES-GCM.

WG(config/policy)#bovpn-tunnel tunnel.1

WG(config/policy/bovpntunnel-tunnel.1)#phase2 proposals ESP-AES256-GCM replace yes

WG(config/policy/bovpntunnel-tunnel.1)#apply

- The Crypto Officer shall not configure MD5, DES or Triple-DES in any and all IPSec VPN configurations in order to operate in an approved mode of operation. Otherwise, the module will be operated in a non-approved mode, which has the log showing non-approved algorithm is being used. For example, the log "fips non-approved mode: using "DES-CBC" cipher" shows non-approved algorithm DEC-CBC is being used in the service implementation.
- When configuring IPSec VPN service, choose Diffie-Hellman with 2048 bits and 3072 bits, EC Diffie-Hellman curves: P-256, P-384 or P-521 elliptic curve.
- When configuring IPSec VPN tunnels, use pre-shared key, RSA certificates, or ECDSA certificates for authentication. Use a minimum of 2048-bits for all RSA keys. Pleases use the Approved authentication and encryption algorithms.
- The Crypto Officer can confirm that all IPSec VPN configurations have been correctly configured by issuing the commands ‘show bovpn-gateway’ and ‘show bovpn-tunnel’ to verify that VPN configuration. Below is an example.

WG#show bovpn-gateway

--

--Phase1 Transform

--

Authentication	*Encryption*	*SA Life*	*Key Group*
SHA2-256	*AES*	*24 Hour(s)*	*Diffie-Hellman Group14*

WG#show bovpn-tunnel

--

-- Phase2 Settings

--

Ipsec Proposals:

ESP-AES256-SHA256

Please note that the approved service indictor can be found in the log file with the associated approved algorithms being used in the service implementation. For example, the log shows "fips approved mode: using "aes-256-cbc" cipher" to denote the module is using approved AES-CBC mode with key size of 256 bits.

- To make the module operate in Bypass state, the CO needs to complete the following configurations.
  - Define the interface property (name, ip address, etc.)

- Create Firewall Polices to define the traffic traversing to destined IP address to be in plaintext
- Apply Firewall Polices to the configuration

For reference, below is an example of Bypass configuration:

WG(config)#policy

WG(config/policy)#rule Allow-Ping

WG(config/policy/rule-Allow-Ping)#policy-type Ping from host-ip < Source IP address> to host-ip <Destined IP Address> firewall allowed

WG(config/policy/rule-Allow-Ping)#logging log-message enable

WG(config/policy/rule-Allow-Ping)#apply

WG(config/policy/rule-Allow-Ping)#exit

WG(config/policy)#rule Allow-TCP-UDP

WG(config/policy/rule-Allow-TCP-UDP)#policy-type TCP-UDP from host-ip <Source IP Address> to host-ip <Destined IP Address> firewall allowed

WG(config/policy/rule-Allow-TCP-UDP)#logging log-message enable

WG(config/policy/rule-Allow-TCP-UDP)#apply

WG(config/policy/rule-Allow-TCP-UDP)#exit

WG(config) #save

- Please refer to section 4.6 above in this document for further description to Bypass service.
- Web browsers must be configured to only use TLS v1.2/v1.3 and approved cipher suites.
- Telnet and SSH clients must be configured to use the SSH v2.0 protocol and RSA authentication. If the SSH client uses Diffie-Hellman key exchange, configure the client to use DH 2048 bit or greater.

11.2 Administrator Guidance

No specific Administrator guidance.

11.3 Non-Administrator Guidance

No specific Non-Administrator guidance.

12 Mitigation of Other Attacks

N/A for this module.