Ruckus Wireless LLC

RUCKUS R770-US Access Point, R770-WW Access Point, R670-US Access Point, R670-WW Access Point, T670-US Access Point, T670-WW Access Point and T670sn-US Access Point

FIPS 140-3 Non-Proprietary Security Policy

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

RUCKUS WiFi 7 access point portfolio comprises of indoor and outdoor APs that brings higher capacity, efficiency, and performance to enterprises, distributed offices, and small businesses. Ruckus WiFi 7 portfolio offers dual-band, dual-concurrent APs that support up to eight spatial streams and tri-band, tri-concurrent APs that support up to eight spatial streams—eventually twelve spatial streams on future models. Equipped with OFDMA and MU-MIMO capabilities, the APs efficiently manage up to 1024 client connections with increased capacity, improved coverage and performance in ultra-high dense environments.

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	2
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:

This is Ruckus Wireless LLC. non-proprietary security policy for RUCKUS R670-US Access Point, R670-WW Access Point, R770-US Access Point, R770-WW Access Point, T670-US Access Point, T670-WW Access Point and T670sn-US Access Point (hereinafter referred to as Module), version 7.1.1.3. 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 module is operated in a limited operational environment.

The Module provides the connection point between wireless client hosts and the wired network. Once authenticated as authenticated nodes on the wired infrastructure, the Module provides the encryption service on the wireless network between themselves and the wireless client via the 802.11i secure service. Then, the Module tunnels the wireless client data via IPsec/IKEv2 to the RUCKUS Data Plane. The Module also communicates directly with the wireless controller for

management purposes. The management traffic between the Module and RUCKUS Wireless Controller is protected by SSHv2 secure tunnel.

Module Type: Hardware

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The 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 the Physical Security section. The US and WW Access Point versions share the same exterior appearance. Where they differ is in the allowed frequencies and power levels allowed in the regulatory domain they are configured to operate in.

Figure 1: RUCKUS R670-US Access Point, R670-WW Access Point

Figure 2: RUCKUS R770-US Access Point, R770-WW Access Point

Figure 3: RUCKUS T670-US Access Point, T670-WW Access Point

Figure 4: RUCKUS T670sn Access Point

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
R670-US Access Point	9F1-R670-US00	7.1.1.3	QCA IPQ5332	
R670-WW Access Point	9F1-R670-WW00	7.1.1.3	QCA IPQ5332	
R770-US Access Point	9F1-R770-US00	7.1.1.3	QCA IPQ9570	
R770-WW Access Point	9F1-R770-WW00	7.1.1.3	QCA IPQ9570	
T670-US Access Point	9F1-T670-US01	7.1.1.3	QCA IPQ5332	
T670-WW Access Point	9F1-T670-WW00	7.1.1.3	QCA IPQ5332	
T670sn-US Access Point	9F1-T670-US51	7.1.1.3	QCA IPQ5332	

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 of Operation	The module is always in the approved mode of operation after initial operations are performed.	Approved	Approved mode indicator: "FIPS mode is enabled."

Table 3: Modes List and Description

After the Module is configured in the Approved mode of operation as outlined in Section 11 of this document, it will be ready to operate in the Approved mode. The Module does not claim the implementation of a degraded mode operation.

2.5 Algorithms

Approved Algorithms:

Ruckus OpenSSL Crypto Implementation

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A6768	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A6768	Key Length - 128	SP 800-38A
AES-CTR	A6768	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-GCM	A6768	Key Length - 128, 256	SP 800-38D
Counter DRBG	A6768	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigGen (FIPS186-5)	A6768	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A6768	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC-SHA-1	A6768	Key Length - Key Length: 160	FIPS 198-1
HMAC-SHA2-256	A6768	Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-512	A6768	Key Length - Key Length: 512	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A6768	Domain Parameter Generation Methods - P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A6768	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, MODP-2048, MODP-3072, MODP-6144, MODP-8192 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDF IKEv2 (CVL)	A6768	Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 2048-8192 Increment 8 Derived Keying Material Length - Derived Keying Material Length: 1056-3072 Increment 8 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
KDF SNMP (CVL)	A6768	Password Length - Password Length: 64, 128	SP 800-135 Rev. 1
KDF SP800-108	A6768	KDF Mode - Counter Supported Lengths - Supported Lengths: 256	SP 800-108 Rev. 1
KDF SSH (CVL)	A6768	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
RSA KeyGen (FIPS186-5)	A6768	Modulo - 3072	FIPS 186-5
RSA SigGen (FIPS186-5)	A6768	Modulo - 3072	FIPS 186-5
RSA SigVer (FIPS186-5)	A6768	Modulo - 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
Safe Primes Key Generation	A6768	Safe Prime Groups - ffdhe2048, ffdhe3072, MODP-2048, MODP-3072, MODP-6144, MODP-8192	SP 800-56A Rev. 3
SHA-1	A6768	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-256	A6768	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A6768	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A6768	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A6768	Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A6768	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE	SP 800-135 Rev. 1

Table 4: Approved Algorithms - Ruckus OpenSSL Crypto Implementation

Ruckus Kernel Crypto Implementation

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A6769	Direction - Decrypt, Encrypt Key Length - 128, 192	SP 800-38A
HMAC-SHA2-256	A6769	Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-384	A6769	Key Length - Key Length: 384	FIPS 198-1
HMAC-SHA2-512	A6769	Key Length - Key Length: 512	FIPS 198-1
SHA2-256	A6769	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A6769	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A6769	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 5: Approved Algorithms - Ruckus Kernel Crypto Implementation

Qualcomm Wi-Fi 7 Hardware AES Engine for Ruckus R670-US/WW, T670-US/WW, T670sn

Algorithm	CAVP Cert	Properties	Reference
AES-CCM	A3235	Key Length - 128, 256	SP 800-38C
AES-GCM	A3235	Direction - Decrypt, Encrypt IV Generation - External IV Generation Mode - 8.2.1 Key Length - 128, 256	SP 800-38D

Table 6: Approved Algorithms - Qualcomm Wi-Fi 7 Hardware AES Engine for Ruckus R670-US/WW, T670-US/WW, T670sn

Qualcomm Wi-Fi 7 Hardware AES Engine for Ruckus T770-US/WW

Algorithm	CAVP Cert	Properties	Reference
AES-CCM	A3239	Key Length - 128, 256	SP 800-38C
AES-GCM	A3239	Direction - Decrypt, Encrypt IV Generation - External IV Generation Mode - 8.2.1 Key Length - 128, 256	SP 800-38D

Table 7: Approved Algorithms - Qualcomm Wi-Fi 7 Hardware AES Engine for Ruckus T770-US/WW

Ruckus Access Point TPM

Algorithm	CAVP Cert	Properties	Reference
RSA SigGen (FIPS186-5)	A8586	Modulo - 3072 Signature Type - pkcs1v1.5	FIPS 186-5

Table 8: Approved Algorithms - Ruckus Access Point TPM

As the Module can only be operated in the Approved mode of operation, and any algorithms not listed in the tables above will be rejected by the module while in the approved mode, the options defined in SP 800-140B for the following categories are missing from this document.

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 section 4 example 1 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 (A6768)

Table 9: 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:

N/A for this module.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
KAS-ECC (SSHv2)	CKG KAS-Full	Full KAS-ECC Key Agreement used for SSHv2 service	Caveat:Key establishment methodology provides between 128 and 256 bits of security strength IG : IG D.F Scenario 2, Path 2, Split Key Confirmation :	KAS-ECC-SSC Sp800-56Ar3: (A6768) Curves: P-256, P-384, P-521 KDF SSH: (A6768) Counter DRBG: (A6768) CKG: ()

Name	Type	Description	Properties	Algorithms
			No Key Derivation : IG 2.4.B SP 800-135rev1 CVL	Key Type: Asymmetric
KAS-ECC (TLSv1.2)	CKG KAS-Full	Full KAS-ECC Key Agreement used for TLSv1.2 service	Caveat:Key establishment methodology provides 128 bits of security strength IG : IG D.F Scenario 2, Path 2, Split Key Confirmation : No Key Derivation : IG 2.4.B SP 800-135rev1 CVL	KAS-ECC-SSC Sp800-56Ar3: (A6768) Curve: P-256 TLS v1.2 KDF RFC7627: (A6768) Counter DRBG: (A6768) CKG: () Key Type: Asymmetric
KAS-ECC (TLSv1.3)	CKG KAS-Full	Full KAS-ECC Key Agreement used for TLSv1.3 service	Caveat:Key establishment methodology provides 128 bits of security strength IG : IG D.F Scenario 2, Path 2, Split Key Confirmation : No Key Derivation : IG 2.4.B SP 800-135rev1 CVL	KAS-ECC-SSC Sp800-56Ar3: (A6768) Curve: P-256 TLS v1.3 KDF: (A6768) Counter DRBG: (A6768) CKG: () Key Type: Asymmetric
KAS-ECC (IKEv2)	CKG KAS-Full	Full KAS-ECC Key Agreement used for IKEv2 service	Caveat:Key establishment methodology provides 192 bits of security strength IG : IG D.F Scenario 2, Path 2, Split Key Confirmation : No Key Derivation : IG 2.4.B SP 800-135rev1 CVL	KAS-ECC-SSC Sp800-56Ar3: (A6768) Curve: P-384 KDF IKEv2: (A6768) Counter DRBG: (A6768) CKG: () Key Type: Asymmetric
KAS-FFC (IKEv2)	CKG KAS-Full	Full KAS-FFC Key Agreement used for IKEv2 service	Caveat: Key establishment methodology provides between 112 and 200 bits of	KAS-FFC-SSC Sp800-56Ar3: (A6768) Domain Parameter Generation:

Name	Type	Description	Properties	Algorithms
			security strength IG : IG D.F Path 2, Scenario 2, Split Key Confirmation : No Key Derivation : IG 2.4.B SP 800-135rev1 CVL	MODP-2048, MODP-3072, MODP6144, MODP8192 Safe Primes Key Generation: (A6768) KDF IKEv2: (A6768) Counter DRBG: (A6768) CKG: () Key Type: Asymmetric
KTS (SSHv2 with AES and HMAC)	KTS-Unwrap	KTS via SSHv2 service by using AES and HMAC	Standard:SP 800-38F IG D.G:Approved "Combination" method Caveat:Key establishment methodology provides 128 or 256 bits of security strength	AES-CTR: (A6768) Key Length: 128, 256 HMAC-SHA2-256: (A6768) HMAC-SHA2-512: (A6768) SHA2-256: (A6768) SHA2-512: (A6768)
KTS (SSHv2 with AES-GCM)	KTS-Unwrap	KTS via SSHv2 service by using AES-GCM	Standard:SP 800-38F IG D.G:Approved "Authenticated" method Caveat:Key establishment methodology provides 128 or 256 bits of security strength	AES-GCM: (A6768) Key Length: 128, 256
RSA KeyGen (TLSv1.2 and TLSv1.3)	AsymKeyPair-KeyGen	RSA KeyGen for TLSv1.2 and TLSv1.3 services		RSA KeyGen (FIPS186-5): (A6768) Modulus: 3072 Counter DRBG: (A6768) CKG: () Key Type: Asymmetric
RSA SigGen (SSHv2,	DigSig-SigGen	RSA SigGen for SSHv2,		RSA SigGen (FIPS186-5):

Name	Type	Description	Properties	Algorithms
TLSv1.2, TLSv1.3, and IPsec/IKEv2)		TLSv1.2, TLSv1.3, and IPsec/IKEv2 services		(A6768) Modulus: 3072
ECDSA SigGen (SSHv2 and IPsec/IKEv2)	DigSig-SigGen	ECDSA SigGen for SSHv2 and IPsec/IKEv2 services		ECDSA SigGen (FIPS186-5): (A6768) Curve: P-384
ECDSA SigGen (TLSv1.2 and TLSv1.3)	DigSig-SigGen	ECDSA SigGen for TLSv1.2 and TLSv1.3 services		ECDSA SigGen (FIPS186-5): (A6768) Curves: P-256, P-384
RSA SigVer (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2)	DigSig-SigVer	RSA SigVer for SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2 services		RSA SigVer (FIPS186-5): (A6768) Modulus: 3072
ECDSA SigVer (SSHv2 and IPsec/IKEv2)	DigSig-SigVer	ECDSA SigVer for SSHv2 and IPsec/IKEv2 services		ECDSA SigVer (FIPS186-5): (A6768) Curve: P-384
ECDSA SigVer (TLSv1.2 and TLSv1.3)	DigSig-SigVer	ECDSA SigGen for TLSv1.2 and TLSv1.3 services		ECDSA SigVer (FIPS186-5): (A6768) Curve: P-256, P-384
SSHv2 Session Encrypt/Decrypt	BC-Auth BC-UnAuth	SSHv2 session protection.		AES-CTR: (A6768) Key Length: 128, 256 AES-GCM: (A6768) Key Length: 128, 256
SSHv2 Session Authentication	MAC	SSHv2 Session Authentication		HMAC-SHA2-256: (A6768) HMAC-SHA2-512: (A6768) SHA2-256: (A6768) SHA2-512: (A6768)
SSHv2 Keying Materials Development	KAS-135KDF	SSHv2 session keying materials, used to derive SSHv2 session keys		KDF SSH: (A6768)

Name	Type	Description	Properties	Algorithms
TLSv1.2 Session Encrypt/Decrypt	BC-Auth	TLSv1.2 session protection.		AES-GCM: (A6768) Key Length: 128, 256
TLSv1.2 Keying Materials Development	KAS-135KDF	TLSv1.2 session keying materials, used to derive TLS session keys		TLS v1.2 KDF RFC7627: (A6768)
TLSv1.3 Session Encrypt/Decrypt	BC-Auth	TLSv1.3 session protection		AES-GCM: (A6768) Key Length: 128, 256
TLSv1.3 Keying Materials Development	KAS-135KDF	TLSv1.3 session keying materials, used to derive TLS session keys		TLS v1.3 KDF: (A6768)
IPsec/IKEv2 Session Encrypt/Decrypt	BC-Auth BC-UnAuth	IPsec/IKEv2 session protection		AES-CBC: (A6768, A6769) Key Length: 128, 192, 256 AES-GCM: (A6768) Key Length: 128, 256
IPsec/IKEv2 Session Authentication	MAC	IPsec/IKEv2 session authentication		HMAC-SHA2-256: (A6769) HMAC-SHA2-384: (A6769) HMAC-SHA2-512: (A6769) SHA2-256: (A6769) SHA2-384: (A6769) SHA2-512: (A6769)
IPsec/IKEv2 Keying Materials Development	KAS-135KDF	IPsec/IKEv2 session keying materials, used to derive IPsec/IKEv2 session keys		KDF IKEv2: (A6768)
SNMPv3 Session Encrypt/Decrypt	BC-UnAuth	SNMPv3 session protection		AES-CFB128: (A6768) Key Length: 128

Name	Type	Description	Properties	Algorithms
SNMPv3 Session Authentication	MAC	Message Authentication for SNMPv3 service		HMAC-SHA-1: (A6768) SHA-1: (A6768)
SNMPv3 Keying Materials Development	KAS-135KDF	SNMPv3 session keying materials, used to derive SNMPv3 session keys		KDF SNMP: (A6768)
802.11i Session Encrypt/ Decrypt	BC-Auth	802.11i Session Protection		AES-CCM: (A3235, A3239) Key Length: 128 AES-GCM: (A3235, A3239) Key Length: 128, 256
802.11i Keying Materials Development	KBKDF	802.11i session keying materials, used to derive 802.11i session keys		KDF SP800-108: (A6768) HMAC-SHA-1: (A6768) HMAC-SHA2-256: (A6768) SHA-1: (A6768) SHA2-256: (A6768)
Firmware Load Test	DigSig-SigVer	Digital signature verification for firmware load test		RSA SigVer (FIPS186-5): (A6768) Modulus: 4096 SHA2-384: (A6768)
DRBG Function	DRBG	Used for DRBG generation		Counter DRBG: (A6768)
Generate Attestation Signature	DigSig-SigGen	Used during initialization of the module to attest to the Controller		RSA SigGen (FIPS186-5): (A8586) Modulus: 3072

Table 10: Security Function Implementations

2.7 Algorithm Specific Information

- For SSHv2, the Module's AES-GCM implementation conforms to Implementation Guidance C.H scenario #1 of FIPS 140-3 IG C.H.

- For TLSv1.2, 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 TLSv1.3, the Module offers the AES-GCM implementation and uses the context of Scenario #1 of FIPS 140-3 IG C.H. The protocol that provides this compliance is TLSv1.3, defined in RFC8446 of August 2018, using the ciphersuites that explicitly select AES-GCM as the encryption/decryption cipher (Appendix B.4 of RFC8446). The Module supports acceptable AES-GCM ciphersuites from Section 3.3.1 of SP800-52rev2. The Module implements, within its boundary, an IV generation unit for TLSv1.3 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.

- For 802.11i, the Module's AES-GCM implementation conforms to Implementation Guidance C.H scenario #3 of FIPS 140-3 IG C.H.

- In accordance with FIPS 140-3 IG D.H, the cryptographic module performs Cryptographic Key Generation as per section 5 in SP800-133rev2. The resulting generated seed used in the asymmetric key generation is the unmodified output from SP800-90Arev1 DRBG.

2.8 RBG and Entropy

Cert Number	Vendor Name
E270	CommScope

Table 11: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Ruckus CPU Jitter Entropy Source	Non-Physical	QCA IPQ5332, QCA IPQ9570	256	Full entropy	A6935 (SHA3-256)

Table 12: Entropy Sources

2.9 Key Generation

The module implements Cryptographic Key Generation (CKG, vendor affirmed), compliant with SP 800- 133r2. When random values are required, they are obtained from the SP 800-90Ar1 approved DRBG, compliant with Section 4 of SP 800-133r2. The following methods are implemented:

- Direct generation of symmetric keys: compliant with SP 800-133rev2, Section 6.1.
- Safe primes key pair generation: compliant with SP 800-133rev2, Section 5.2, which maps to SP 800-56Arev3. The method described in Section 5.6.1.1.4 of SP 800-56Ar3 (“Testing Candidates”) is used.
- RSA key pair generation: compliant with SP 800-133rev2, Section 5.1, which maps to FIPS 186-5. The method described in Appendix A.1.3 of FIPS 186-5 (“Probable Primes”) is used.
- ECDH key pair generation: compliant with SP 800-133r2, Section 5.1, which maps to FIPS 186-5. The method described in Appendix A.2.2 of FIPS 186-5 (“Testing Candidates”) is used. Note that this generation method is also used to only generate ECDH key pairs.

Additionally, the module implements the following key derivation methods:

- SSHv2 KDF, TLS 1.2 KDF (RFC 7627), TLSv1.3 KDF, IKEv2 KDF and SNMPv3 KDF: compliant with SP 800-135r1. These implementations shall only be used to generate secret keys in the context of the SSHv2, TLSv1.2, TLSv1.3, IKEv2 and SNMPv3 protocols, respectively.
- KDF SP800-108: compliant with section 4.1 “KDF in Counter Mode” of SP 800-108rev1. This implementation shall only be used in context of the 802.11i protocol.

Intermediate key generation values are not output from the module and are explicitly zeroized after processing the service

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 between 112 and 200 bits of encryption strength.

- The module supports the use of the safe primes defined in RFC 3526 (IKE).
- Note that the module only implements domain parameter generation, key pair generation and verification, and shared secret computation:
    - IKE (RFC 3526):
        - MODP-2048 (ID = 14)
        - MODP-3072 (ID = 15)
        - MODP-6144 (ID = 17)
        - MODP-8192 (ID = 18)

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, IPsec/IKEv2, SNMPv3 and 802.11i industrial protocols. No parts of SSHv2, TLSv1.2, TLSv1.3, IPSec/IKEv2, SNMPv3, and 802.11i 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
Ethernet Port and RF Interface	Data Input	Data input into the module for all the services defined in Approved Services Table, including SSHv2, TLSv1.2, TLSv1.3, IPsec/IKEv2and SNMPv3 service data.
Ethernet Port and RF Interface	Data Output	Data output from the module for all the services defined in Approved Services Table, including SSHv2, TLSv1.2, TLSv1.3, IPsec/IKEv2and SNMPv3 service data.
Ethernet Port and RESET	Control Input	Control Data input into the module for all the services defined in Approved Services Table, including SSHv2, TLSv1.2, TLSv1.3, IPsec/IKEv2and SNMPv3 service data.
Ethernet Port and LED	Status Output	Status Information output from the module.
N/A	Control Output	N/A
Power	Power	Provide the Power Supply to the module.

Table 13: 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	The minimum length is fifteen (15) characters (94 possible characters).	Password Based	The probability that a random attempt will succeed or a false acceptance will occur is 1/(94^15) which is less than 1/1,000,000.	The probability of successfully authenticating to the module within one minute is 10/(94^15), which is less than 1/100,000.
RSA-Based Certificate	The modules support RSA public-key based authentication mechanism using a minimum of RSA 3072 bits, 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 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.	RSA SigVer (FIPS186-5) (A6768)	The probability that a random attempt will succeed is 1/(2^128). 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
ECDSA-Based Certificate	The modules support ECDSA public-key based authentication	ECDSA SigVer	The probability that a random attempt will	The probability of successfully authenticating to the

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	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 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.	(FIPS186-5) (A6768)	succeed is 1/(2^128) which is less than 1/1,000,000. Please refer to Description section in this table for more details	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

Table 14: Authentication Methods

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	Password RSA-Based Certificate ECDSA-Based Certificate
User (Wireless Client)	Role	User	Password

Table 15: Roles

The Module supports Crypto Officer (CO) role and User role (Wireless Client).

4.3 Approved Services

The following tables detail the types of approved services available to each role in approved mode of operation, the types of access for each role and the Keys or SSPs they affect.

- Generate G
- Read Access R
- Write Access W

- Execute Access E
- Zeroize Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show Status	Provide Module's current status (return codes and/or syslog messages)	N/A	Commands used to show Module's Status	Module's Operational Status	None	Crypto Officer
Show Version	Provide Module's name and version information	N/A	Commands `get boarddata` and `get fips-version`	Module's ID and versioning information	None	Crypto Officer
Perform Self-Tests	Perform Self-Tests (Pre-operational self-test and Conditional Self-Tests)	N/A	Command to trigger Self-Test	Status of the self-tests results	None	Crypto Officer Unauthenticated
Perform Zeroization	Perform Zeroization	Syslog message	Command to zeroize the module	Status of the SSPs zeroization	None	Crypto Officer - DRBG Entropy Input: Z - DRBG Seed: Z - DRBG Internal State V value: Z - DRBG Key: Z - Crypto Officer Password: Z - Firmware Load Test Key: Z - SSHv2 ECDH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Private Key: Z - SSHv2 ECDH Public Key: Z - SSHv2 Peer ECDH Public Key: Z - SSHv2 ECDH Shared Secret: Z - SSHv2 RSA Private Key: Z - SSHv2 RSA Public Key: Z - SSHv2 RSA Peer Public Key: Z - SSHv2 ECDSA Private Key: Z - SSHv2 ECDSA Public Key: Z - SSHv2 ECDSA Peer Public Key: Z - SSHv2 Session Encryption Key: Z - SSHv2 Session Authentication Key: Z - TLSv1.2 ECDH Private Key: Z - TLSv1.2

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						ECDH Public Key: Z - TLSv1.2 Peer ECDH Public Key: Z - TLSv1.2 ECDH Shared Secret: Z - Locally generated TLSv1.2 RSA Private Key: Z - Locally generated TLSv1.2 RSA Public Key: Z - TLSv1.2 RSA Private Key: Z - TLSv1.2 RSA Public Key: Z - TLSv1.2 RSA Peer Public Key: Z - TLSv1.2 ECDSA Private Key: Z - TLSv1.2 ECDSA Public Key: Z - TLSv1.2 ECDSA Peer Public Key: Z - TLSv1.2 Master Secret: Z - TLSv1.2 Session Encryption

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key: Z - TLSv1.3 ECDH Private Key: Z - TLSv1.3 ECDH Public Key: Z - TLSv1.3 Peer ECDH Public Key: Z - TLSv1.3 ECDH Shared Secret: Z - Locally generated TLSv1.3 RSA Private Key: Z - Locally generated TLSv1.3 RSA Public Key: Z - TLSv1.3 RSA Private Key: Z - TLSv1.3 RSA Public Key: Z - TLSv1.3 RSA Peer Public Key: Z - TLSv1.3 ECDSA Private Key: Z - TLSv1.3 ECDSA Public Key: Z - TLSv1.3 ECDSA Peer Public Key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLSv1.3 Master Secret: Z - TLSv1.3 Session Encryption Key: Z - IPsec/IKEv2 DH Private Key: Z - IPsec/IKEv2 DH Public Key: Z - IPsec/IKEv2 Peer DH Public Key: Z - IPsec/IKEv2 DH Shared Secret: Z - IPsec/IKEv2 ECDH Private Key: Z - IPsec/IKEv2 ECDH Public Key: Z - IPsec/IKEv2 Peer ECDH Public Key: Z - IPsec/IKEv2 ECDH Shared Secret: Z - IPsec/IKEv2 RSA Private Key: Z -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						IPsec/IKEv2 RSA Public Key: Z - IPsec/IKEv2 RSA Peer Public Key: Z - IPsec/IKEv2 ECDSA Private Key: Z - IPsec/IKEv2 ECDSA Public Key: Z - IPsec/IKEv2 ECDSA Peer Public Key: Z - IPsec/IKEv2 Pre-shared Secret: Z - SKEYSEED: Z - IPsec/IKEv2 Session Encryption Key: Z - IPsec/IKEv2 Authentication Key: Z - SNMPv3 Shared Secret: Z - SNMPv3 Encryption Key: Z - SNMPv3 Authentication Key: Z - 802.11i

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Pre-Shared Secret: Z - 802.11i Pairwise Master Key (PMK): Z - 802.11i Pairwise Transient Key (PTK): Z - 802.11i Temporal Key (TK): Z - 802.11i Group Master Key (GMK): Z - 802.11i Group Temporal Key (GTK): Z
Configure Network	Sets configuration of the systems	None	Commands to configure the network	Status of the completion of network configuration status	None	Crypto Officer
Crypto Officer Authentication	CO Role Authentication	N/A	CO Authentication Request	Status of the CO authentication	None	Crypto Officer - Crypto Officer Password: W - SSHv2 RSA Peer Public Key: W,E - SSHv2 ECDSA Peer Public Key: W,E
User (Wireless Client) Authentication	User (Wireless Client) Role	N/A	User (Wireless Client) Authentica	Status of the User (Wireless Client)	None	User (Wireless Client) - User (Wireless

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	Authentication		tion Request	Authentication		Client) Password: W
Initialization Attestation	The module will generate a signature to attest to the Controller during initialization	The module is successfully connected to the controller	Commands to connect to the controller	Outcome of the attestation attempt	Generate Attestation Signature	Crypto Officer - Factory Device Certificate: E
Configure SSHv2 Function	Configure SSHv2 Function	Status Mode indicator "FIPS mode is enabled." and SSHv2 configuration success status message	Commands to configure SSHv2	Status of the completion of the SSHv2 configuration	KTS (SSHv2 with AES and HMAC) KTS (SSHv2 with AES-GCM)	Crypto Officer - SSHv2 RSA Private Key: W - SSHv2 RSA Public Key: W - SSHv2 ECDSA Private Key: W - SSHv2 ECDSA Public Key: W
Configure TLSv1.2 Function	Configure TLSv1.2 Function	Status Mode indicator "FIPS mode is enabled." and TLSv1.2 configuration success status message	Commands to configure TLSv1.2	Status of the completion of TLSv1.2 configuration	KTS (SSHv2 with AES and HMAC) KTS (SSHv2 with AES-GCM) RSA KeyGen (TLSv1.2 and TLSv1.3) DRBG Function	Crypto Officer - Locally generated TLSv1.2 RSA Private Key: G,W - Locally generated TLSv1.2 RSA Public Key: G,W - TLSv1.2 RSA Private Key: W - TLSv1.2 RSA Public Key: W - TLSv1.2

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						ECDSA Private Key: W - TLSv1.2 ECDSA Public Key: W - DRBG Entropy Input: G,W,E - DRBG Seed: G,W,E - DRBG Internal State V value: G,W,Z - DRBG Key: G,W,E
Configure TLSv1.3 Function	Configure TLSv1.3 Function.	Status Mode indicator "FIPS mode is enabled." and TLSv1.3 configuration success status message.	API commands to configur TLSv1.3	Status of the completion of TLSv1.3 configuration.	KTS (SSHv2 with AES and HMAC) KTS (SSHv2 with AES-GCM) RSA KeyGen (TLSv1.2 and TLSv1.3) DRBG Function	Crypto Officer - Locally generated TLSv1.3 RSA Private Key: G,W - Locally generated TLSv1.3 RSA Public Key: G,W - TLSv1.3 RSA Private Key: W - TLSv1.3 RSA Public Key: W - TLSv1.3 ECDSA Private Key: W - TLSv1.3 ECDSA Public Key: W - DRBG Entropy

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Input: G,W,E - DRBG Seed: G,W,E - DRBG Internal State V value: G,W,E - DRBG Key: G,W,E
Configure IPsec/IKEv2 Function	Configure IPsec/IKEv2 Function	Status Mode indicator "FIPS mode is enabled." and IPsec/IKEv2 configuration success status message	Commands to configure IPsec/IKEv2	Status of the completion of IPsec/IKEv2 configuration	KTS (SSHv2 with AES and HMAC) KTS (SSHv2 with AES-GCM)	Crypto Officer - IPsec/IKEv2 RSA Private Key: W - IPsec/IKEv2 RSA Public Key: W - IPsec/IKEv2 ECDSA Private Key: W - IPsec/IKEv2 ECDSA Public Key: W - IPsec/IKEv2 Pre-shared Secret: W
Configure SNMPv3 Function	Configure SNMPv3 Function	Status Mode indicator "FIPS mode is enabled." and SNMPv3 configuration success status message	Commands to configure SNMPv3	Status of the completion of SNMPv3 configuration	KTS (SSHv2 with AES and HMAC) KTS (SSHv2 with AES-GCM) SNMPv3 Keying Materials Development	Crypto Officer - SNMPv3 Shared Secret: W - SNMPv3 Encryption Key: G,W - SNMPv3 Authentication Key: G,W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure 802.11i Function	Configure 802.11i Function	Status Mode indicator "FIPS mode is enabled." and 802.11i configuration success status message	API commands to configure 802.11i	API commands to configure 802.11i	KTS (SSHv2 with AES and HMAC) KTS (SSHv2 with AES-GCM)	Crypto Officer - 802.11i Pre-Shared Secret: W - 802.11i Pairwise Master Key (PMK): G - 802.11i Pairwise Transient Key (PTK): G - 802.11i Temporal Key (TK): G - 802.11i Group Master Key (GMK): G - 802.11i Group Temporal Key (GTK): G
Run SSHv2 Function	Execute SSHv2 Function	Status Mode indicator "FIPS mode is enabled." and successful SSHv2 log message	Initiate SSHv2 tunnel establishment	Status of SSHv2 tunnel establishment	KAS-ECC (SSHv2) RSA SigGen (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2) ECDSA SigGen (SSHv2 and IPsec/IKEv2) RSA SigVer (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2) ECDSA	Crypto Officer - SSHv2 ECDH Private Key: G,W,E - SSHv2 ECDH Public Key: G,R,W - SSHv2 Peer ECDH Public Key: W,E - SSHv2 ECDH Shared Secret: G,W,E - SSHv2 RSA Private Key: W,E - SSHv2

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					SigVer (SSHv2 and IPsec/IKEv2) SSHv2 Session Encrypt/Decrypt SSHv2 Session Authentication SSHv2 Keying Materials Development DRBG Function	RSA Public Key: R,W - SSHv2 RSA Peer Public Key: R,W,E - SSHv2 ECDSA Private Key: W,E - SSHv2 ECDSA Public Key: R,W - SSHv2 ECDSA Peer Public Key: R,W,E - SSHv2 Session Encryption Key: G,W,E - SSHv2 Session Authentication Key: G,W,E - DRBG Entropy Input: G,W,E - DRBG Seed: G,W,E - DRBG Internal State V value: G,W,E - DRBG Key: G,W,E
Run TLSv1.2 Function	Execute TLSv1.2 function	Status Mode indicator "FIPS mode is enabled." and successfu	Initiate TLSv1.2 tunnel establishment request	Status of TLSv1.2 tunnel establishment	KAS-ECC (TLSv1.2) RSA SigGen (SSHv2, TLSv1.2, TLSv1.3, and	Crypto Officer - TLSv1.2 ECDH Private Key: G,W,E - TLSv1.2 ECDH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		l TLSv1.2 log message			IPsec/IKEv2) ECDSA SigGen (TLSv1.2 and TLSv1.3) RSA SigVer (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2) ECDSA SigVer (TLSv1.2 and TLSv1.3) TLSv1.2 Session Encrypt/Decrypt TLSv1.2 Keying Materials Development DRBG Function	Public Key: G,R,W - TLSv1.2 Peer ECDH Public Key: W,E - TLSv1.2 ECDH Shared Secret: G,E - Locally generated TLSv1.2 RSA Private Key: G,W,E - Locally generated TLSv1.2 RSA Public Key: G,R,W - TLSv1.2 RSA Private Key: W,E - TLSv1.2 RSA Public Key: R,W - TLSv1.2 RSA Peer Public Key: R,W,E - TLSv1.2 ECDSA Private Key: W,E - TLSv1.2 ECDSA Public Key: R,W - TLSv1.2 ECDSA Peer Public Key: R,W,E - TLSv1.2 Master Secret: G,W,E - TLSv1.2 Session Encryption

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key: G,W,E - DRBG Entropy Input: G,W,E - DRBG Seed: G,W,E - DRBG Internal State V value: G,W,E - DRBG Key: G,W,E
Run TLSv1.3 Function	Execute TLSv1.3 Function.	Status Mode indicator "FIPS mode is enabled." and successful TLSv1.3 log message.	API command to execute TLSv1.3 service.	Status of TLSv1.3 establishment.	KAS-ECC (TLSv1.3) RSA SigGen (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2) ECDSA SigGen (TLSv1.2 and TLSv1.3) RSA SigVer (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2) ECDSA SigVer (TLSv1.2 and TLSv1.3) TLSv1.3 Session Encrypt/Decrypt TLSv1.3 Keying Materials	Crypto Officer - TLSv1.3 ECDH Private Key: G,W,E - TLSv1.3 ECDH Public Key: G,R,W - TLSv1.3 Peer ECDH Public Key: W,E - TLSv1.3 ECDH Shared Secret: G,W,E - Locally generated TLSv1.3 RSA Private Key: G,W,E - Locally generated TLSv1.3 RSA Public Key: G,R,W - TLSv1.3 RSA Private Key: W,E - TLSv1.3 RSA Public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Development DRBG Function	Key: R,W - TLSv1.3 RSA Peer Public Key: R,W,E - TLSv1.3 ECDSA Private Key: W,E - TLSv1.3 ECDSA Public Key: R,W - TLSv1.3 ECDSA Peer Public Key: R,W,E - TLSv1.3 Master Secret: G,W,E - TLSv1.3 Session Encryption Key: G,W,E - DRBG Entropy Input: G,W,E - DRBG Seed: G,W,E - DRBG Internal State V value: G,W,E - DRBG Key: G,W,E
Run IPsec/IKEv2 Function	Execute IPsec/IKEv2 Function	Status Mode indicator "FIPS mode is enabled." and succesful IPsec/IKE	Initiate IPsec/IKEv2 tunnel establishment request	Status of IPsec/IKEv2 tunnel establishment	KAS-ECC (IKEv2) KAS-FFC (IKEv2) RSA SigGen (SSHv2, TLSv1.2, TLSv1.3, and	Crypto Officer - IPsec/IKEv2 DH Private Key: G,W,E - IPsec/IKEv2 DH Public Key: G,R,W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		v2 log message			IPsec/IKEv2) ECDSA SigGen (SSHv2 and IPsec/IKEv2) RSA SigVer (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2) ECDSA SigVer (SSHv2 and IPsec/IKEv2) IPsec/IKEv2 Session Encrypt/Decrypt IPsec/IKEv2 Session Authentication IPsec/IKEv2 Keying Materials Development DRBG Function	- IPsec/IKEv2 Peer DH Public Key: W,E - IPsec/IKEv2 DH Shared Secret: G,W,E - IPsec/IKEv2 ECDH Private Key: G,W,E - IPsec/IKEv2 ECDH Public Key: G,R,W - IPsec/IKEv2 Peer ECDH Public Key: W,E - IPsec/IKEv2 ECDH Shared Secret: G,W,E - IPsec/IKEv2 RSA Private Key: W,E - IPsec/IKEv2 RSA Public Key: R,W - IPsec/IKEv2 RSA Peer Public Key: R,W,E - IPsec/IKEv2 ECDSA Private Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- IPsec/IKEv2 ECDSA Public Key: R,W - IPsec/IKEv2 ECDSA Peer Public Key: R,W,E - IPsec/IKEv2 Pre-shared Secret: G,W,E - SKEYSEED: G,W,E - IPsec/IKEv2 Session Encryption Key: G,W,E - IPsec/IKEv2 Authentication Key: G,W,E - DRBG Entropy Input: G,W,E - DRBG Seed: G,W,E - DRBG Internal State V value: G,W,E - DRBG Key: G,W,E
Run SNMPv3 Function	Execute SNMPv3 Function	Status Mode indicator "FIPS mode is enabled." and	Initiate SNMPv3 tunnel establishment request	Status of SNMPv3 tunnel establishment	SNMPv3 Session Encrypt/Decrypt SNMPv3 Session Authenticati	Crypto Officer - SNMPv3 Shared Secret: W,E - SNMPv3 Encryption

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		successful SNMPv3 log message			on SNMPv3 Keying Materials Development	Key: G,W,E - SNMPv3 Authentication Key: G,W,E
Run 802.11i Function	Execute 802.11i Function	Status Mode indicator "FIPS mode is enabled." and successful 802.11i log message	Initiate 802.11i tunnel establishment request	Status of 802.11i tunnel establishment	802.11i Session Encrypt/ Decrypt 802.11i Keying Materials Development	User (Wireless Client) - 802.11i Pre-Shared Secret: G,R,W - 802.11i Pairwise Master Key (PMK): G,R,W,E - 802.11i Pairwise Transient Key (PTK): G,R,W,E - 802.11i Temporal Key (TK): G,R,W,E - 802.11i Group Master Key (GMK): G,R,W,E - 802.11i Group Temporal Key (GTK): G,R,W
Firmware Load Test	Execute the Firmware Load Test	Successful Firmware Loading status message	Commands to load new firmware image	Outcome of the Firmware Load Test	Firmware Load Test	Crypto Officer - Firmware Load Test Key: R,E

Table 16: Approved Services

4.4 Non-Approved Services

N/A for this module.

4.5 External Software/Firmware Loaded

The Module supports the firmware load test by using RSA 4096 SigVer with SHA2-384 (Cert. #A6768) for the new validated firmware to be uploaded into the module. 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. Any firmware 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.

4.6 Bypass Actions and Status

N/A for this module.

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. The unauthenticated operator can trigger 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, all firmware components within the physical boundary are protected by RSA 4096 SigVer with SHA2-384 (Cert. #A6768) signature calculated at build time. At initialization, the signature is recalculated and compared to the hardcoded build-time generated signature value. If at load time the signature does not match, the module exits with an error. If failure occurs during self-test, all crypto functionality is disabled.

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 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 labels	Recommended 30 days	Visible inspection of platform for residual evidence of tampering
Strong Enclosure	Recommended 30 days	Visible inspection of platform for evidence of tampering, removal or access
Production Grade Components	Recommended 30 days	Visible inspection of components for evidence of tampering, removal or access

Table 17: Mechanisms and Actions Required

Appling Tamper Evidence Labels

Step 1: Turn off and unplug the module.

Step 2: Clean the chassis of any grease, dirt, oil or any other material other than the surface coating from manufacture before applying the tamper evident labels. Alcohol-based cleaning pads are recommended for this purpose.

Step 3: Apply a label to cover the Module as shown in the figures below.

The tamper evident labels are produced from a special thin gauge vinyl with self-adhesive backing. Any attempt to open the module will damage the tamper evident labels or the material of the security appliance cover. Because the tamper evident labels have non-repeated serial numbers, they may be inspected for damage and compared against the applied serial numbers to verify that the security appliance has not been tampered with. Tamper evident labels can also be inspected for signs of tampering, which include the following: curled corners, rips, and slices.

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 Approved mode of operation. The Crypto Officer (CO) checks the integrity of the label by following the guidance listed above, any unused TELs must be securely stored, accounted for, and maintained by the Crypto Officer (CO) in a protected location.

7.2 User Placed Tamper Seals

1. RUCKUS R670-US Access Point and R670-WW Access Point

Number: Four (4)

Placement:

Figure 5: R670-US/WW Top

Figure 6: R670-US/WW Bottom

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

2. RUCKUS R770-US Access Point and R770-WW Access Point

Number: Four (4)

Placement:

Figure 7: R770-US/WW Top

Figure 8: R770-US/WW Bottom

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

3. RUCKUS T670-US Access Point and T670-WW Access Point

Number: Four (4)

Placement:

Figure 9: T670-US/WW Top

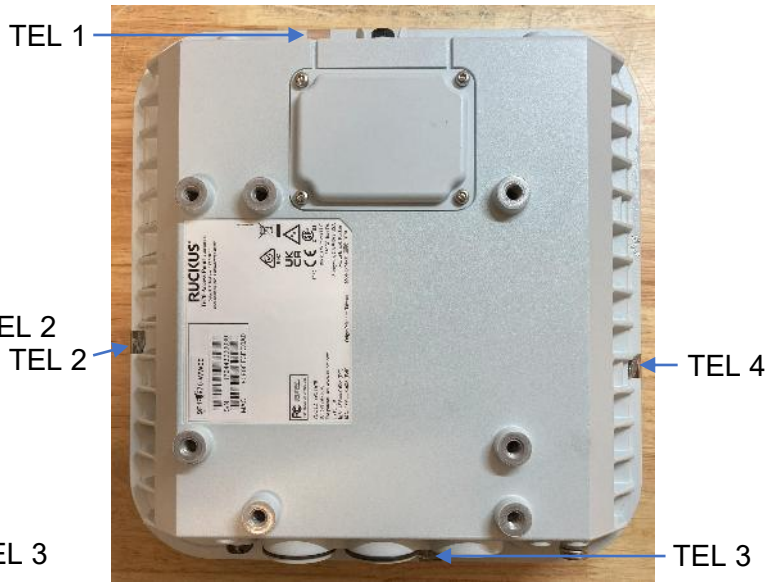

Figure 10: T670-US/WW Bottom

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

4. RUCKUS T670sn Access Point

Number: Four (4)

Placement:

Figure 11: T670sn Top

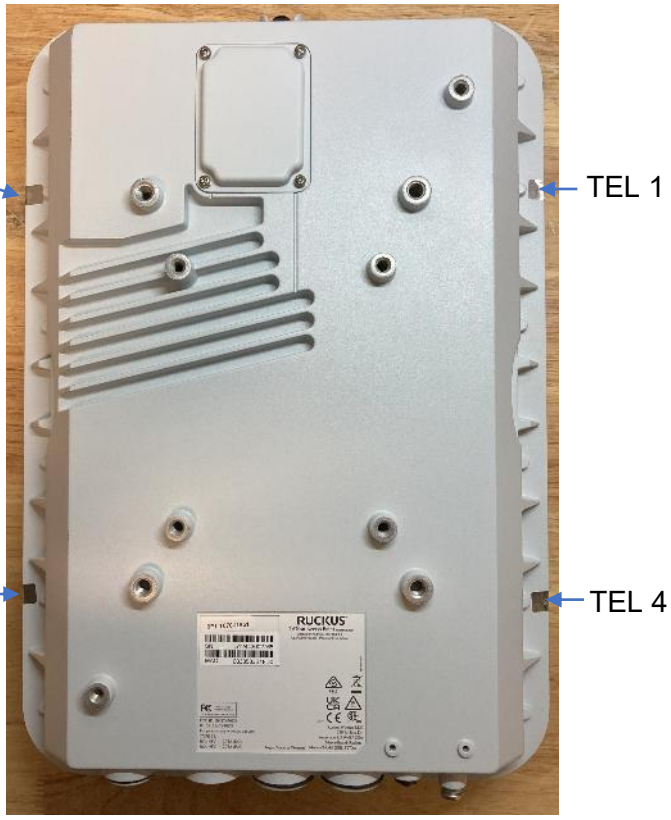

Figure 12: T670sn Bottom

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

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 18: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Peer Public Key Input	External (Outside of the Module's Boundary)	Module	Plaintext	Automated	Electronic	
Module Public Key Output	Module	External (Outside of the Module's Boundary)	Plaintext	Automated	Electronic	
Secret Input via SSHv2 encrypted by AES and HMAC	External (Outside of the Module's Boundary)	Module	Encrypted	Automated	Electronic	KTS (SSHv2 with AES and HMAC)
Secret Input via SSHv2 encrypted by GCM	External (Outside of the Module's Boundary)	Module	Encrypted	Automated	Electronic	KTS (SSHv2 with AES-GCM)

Table 19: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Zeroization Method	CO issues zeroization service via Controller	The zeroization method will erase all SSPs stored in the DRAM or in the Flash of the module.	`zeroize-all csp` or Zeroization method via Controller
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 volatile memory	Reboot

Table 20: SSP Zeroization Methods

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 - at least 256 bits	Entropy Input - CSP			DRBG Function
DRBG Seed	Used in DRBG Generation	256 bits - 256 bits	DRBG Seed - CSP			DRBG Function
DRBG Internal State V value	Used in DRBG Generation	256 bits - 256 bits	DRBG Internal State V value - CSP			DRBG Function
DRBG Key	Used in DRBG Generation	256 bits - 256 bits	DRBG Key - CSP			DRBG Function
Crypto Officer Password	Crypto Officer authentication	8-30 Characters - 8-30 Characters	Authentication Data - CSP			
User (Wireless Client) Password	Used to Authenticate the User (Wireless Client) Role	8-30 Characters - 8-30 Characters	Authentication Data - CSP			
Factory Device Certificate	Used for initialization attestation with Controller for first time,	Modulus 3072 bits - 128 bits	Certificate - CSP			Generate Attestation Signature

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	imported at the factory					
Firmware Load Test Key	Used for Firmware Load Test	4096 bits - 152 bits	Public Key - CSP			Firmware Load Test
SSHv2 ECDH Private Key	Used to derive the SSHv2 ECDH Shared Secret	Curves: P-256, P-384, P-521 - 128 to 256 bits	Private Key - CSP	KAS-ECC (SSHv2)		KAS-ECC (SSHv2)
SSHv2 ECDH Public Key	Used to derive SSHv2 ECDH Shared Secret	Curves: P-256, P-384, P-521 - 128 to 256 bits	Public Key - PSP		KAS-ECC (SSHv2)	
SSHv2 Peer ECDH Public Key	Used to derive SSHv2 DH Shared Secret	Curves: P-256, P-384, P-521 - 128 to 256 bits	Public Key - PSP			KAS-ECC (SSHv2)
SSHv2 ECDH Shared Secret	Used to derive SSHv2 Session Encryption Keys, SSHv2 Session Authentication Keys	Curves: P-256, P-384, P-521 - 128 to 256 bits	Shared Secret - CSP		KAS-ECC (SSHv2)	SSHv2 Keying Materials Development
SSHv2 RSA Private Key	Imported by the controller used for SSHv2 authentication	Modulus 3072 bits - 128 bits	Private Key - CSP			RSA SigGen (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2)
SSHv2 RSA Public Key	Imported by the controller used for SSHv2 authentication	Modulus 3072 bits - 128 bits	Public Key - PSP			

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SSHv2 RSA Peer Public Key	Used for peer authentication during the SSHv2 service	Modulus 3072 bits - 128 bits	Public Key - PSP			RSA SigVer (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2)
SSHv2 ECDSA Private Key	Imported by the controller used for SSHv2 authentication	Curve: P-384 - 192 bits	Private Key - CSP			ECDSA SigGen (SSHv2 and IPsec/IKEv2)
SSHv2 ECDSA Public Key	Imported by the controller used for SSHv2 authentication	Curve: P-384 - 192 bits	Public Key - PSP			
SSHv2 ECDSA Peer Public Key	Used for peer authentication during the SSHv2 service	Curve: P-384 - 192 bits	Public Key - PSP			ECDSA SigVer (SSHv2 and IPsec/IKEv2)
SSHv2 Session Encryption Key	Used for SSHv2 Session protection	128 to 256 bits - 128 to 256 bits	Session Key - CSP		SSHv2 Keying Materials Development	SSHv2 Session Encrypt/Decrypt
SSHv2 Session Authentication Key	Used for SSHv2 Session integrity protection	At least 160 bits - At least 160 bits	Session Key - CSP		SSHv2 Keying Materials Development	SSHv2 Session Authentication
TLSv1.2 ECDH Private Key	Used to Derive TLSv1.2 ECDH Shared Secret	Curve: P-256 - 128 bits	Private Key - CSP	KAS-ECC (TLSv1.2)		KAS-ECC (TLSv1.2)
TLSv1.2 ECDH Public Key	Used to Derive TLSv1.2 ECDH Shared Secret	Curve: P-256 - 128 bits	Public Key - PSP		KAS-ECC (TLSv1.2)	

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TLSv1.2 Peer ECDH Public Key	Used to derive TLSv1.2 ECDH Shared Secret	Curve: P-256 - 128 bits	Public Key - PSP			KAS-ECC (TLSv1.2)
TLSv1.2 ECDH Shared Secret	Used to Derive TLSv1.2 Master Secret and Session Encryption Key	Curve: P-256 - 128 bits	Shared Secret - CSP		KAS-ECC (TLSv1.2)	TLSv1.2 Keying Materials Development
Locally generated TLSv1.2 RSA Private Key	Used for peer authentication during the WISPr captive portal over TLSv1.2	Modulus 3072 bits - 128 bits	Private Key - CSP	RSA KeyGen (TLSv1.2 and TLSv1.3)		RSA SigGen (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2)
Locally generated TLSv1.2 RSA Public Key	Used for peer authentication during the WISPr captive portal over TLSv1.2	Modulus 3072 bits - 128 bits	Public Key - PSP		RSA KeyGen (TLSv1.2 and TLSv1.3)	
TLSv1.2 RSA Private Key	Imported by the controller used for TLSv1.2 authentication	Modulus 3072 bits - 128 bits	Private Key - CSP			RSA SigGen (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2)
TLSv1.2 RSA Public Key	Imported by the controller used for TLSv1.2 authentication	Modulus 3072 bits - 128 bits	Public Key - PSP			
TLSv1.2 RSA Peer Public Key	Used for peer authentication during	Modulus 3072 bits - 128 bits	Public Key - PSP			RSA SigVer (SSHv2, TLSv1.2,

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	the TLSv1.2 service					TLSv1.3, and IPsec/IKEv2)
TLSv1.2 ECDSA Private Key	Imported by the controller used for TLSv1.2 authentication	Curves: P-256, P-384 - 128 or 192 bits	Private Key - CSP			ECDSA SigGen (TLSv1.2 and TLSv1.3)
TLSv1.2 ECDSA Public Key	Imported by the controller used for TLSv1.2 authentication	Curves P-256, P-384 - 128 or 192 bits	Public Key - PSP			
TLSv1.2 ECDSA Peer Public Key	Used for peer authentication during the TLSv1.2 service	Curves: P-256, P-384 - 128 or 192 bits	Public Key - PSP			ECDSA SigVer (TLSv1.2 and TLSv1.3)
TLSv1.2 Master Secret	Used to derive the TLSv1.2 Session Encryption Key	At least 112 bits - At least 112 bits	Master Secret - CSP		TLSv1.2 Keying Materials Development	TLSv1.2 Keying Materials Development
TLSv1.2 Session Encryption Key	Used to protect the TLSv1.2 session	128 or 256 bits - 128 or 256 bits	Session Key - CSP		TLSv1.2 Keying Materials Development	TLSv1.2 Session Encrypt/Decrypt
TLSv1.3 ECDH Private Key	Used to Derive TLSv1.3 ECDH Shared Secret	Curve: P-256 - 128 bits	Private Key - CSP	KAS-ECC (TLSv1.3)		KAS-ECC (TLSv1.3)
TLSv1.3 ECDH Public Key	Used to Derive TLSv1.3 ECDH Shared Secret	Curve: P-256 - 128 bits	Public Key - PSP		KAS-ECC (TLSv1.3)	

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TLSv1.3 Peer ECDH Public Key	Used to derive TLSv1.3 ECDH Shared Secret	Curve: P-256 - 128 bits	Public Key - PSP			KAS-ECC (TLSv1.3)
TLSv1.3 ECDH Shared Secret	Used to Derive TLSv1.3 Session Encryption Key and TLS Session Authentication Key	Curve: P-256 - 128 bits	Shared Secret - CSP		KAS-ECC (TLSv1.3)	TLSv1.3 Keying Materials Development
Locally generated TLSv1.3 RSA Private Key	Used for peer authentication during the WISPr captive portal over TLSv1.3	Modulus 3072 bits - 128 bits	Private Key - CSP	RSA KeyGen (TLSv1.2 and TLSv1.3)		RSA SigGen (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2)
Locally generated TLSv1.3 RSA Public Key	Used for peer authentication during the WISPr captive portal over TLSv1.3	Modulus 3072 bits - 128 bits	Public Key - PSP		RSA KeyGen (TLSv1.2 and TLSv1.3)	
TLSv1.3 RSA Private Key	Imported by the controller used for TLSv1.3 authentication	Modulus 3072 bits - 128 bits	Private Key - CSP			ECDSA SigGen (TLSv1.2 and TLSv1.3)
TLSv1.3 RSA Public Key	Imported by the controller used for TLSv1.3 authentication	Modulus 3072 bits - 128 bits	Public Key - PSP			

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TLSv1.3 RSA Peer Public Key	Used for peer authentication during the TLSv1.3 service	Modulus 3072 bits - 128 bits	Public Key - PSP			ECDSA SigVer (TLSv1.2 and TLSv1.3)
TLSv1.3 ECDSA Private Key	Imported by the controller used for TLSv1.3 authentication	Curves P-256, P-384 - 128 or 192 bits	Private Key - CSP			ECDSA SigGen (TLSv1.2 and TLSv1.3)
TLSv1.3 ECDSA Public Key	Imported by the controller used for TLSv1.3 authentication	Curves P-256, P-384 - 128 or 192 bits	Public Key - PSP			
TLSv1.3 ECDSA Peer Public Key	Used for peer authentication during the TLSv1.3 service	Curves P-256, P-384 - 128 or 192 bits	Public Key - PSP			ECDSA SigVer (TLSv1.2 and TLSv1.3)
TLSv1.3 Master Secret	Used to derive the TLSv1.2 Session Encryption Key	384 bits - 384 bits	Master Secret - CSP		TLSv1.3 Keying Materials Development	TLSv1.3 Session Encrypt/Decrypt
TLSv1.3 Session Encryption Key	Used to protect the TLSv1.3 session	128 or 256 bits - 128 or 256 bits	Symmetric Key - CSP		TLSv1.3 Keying Materials Development	TLSv1.3 Session Encrypt/Decrypt
IPsec/IKEv2 DH Private Key	Used to derive IPsec/IKEv2 DH Shared Secret	MODP-2048, MODP-3072, MODP6144, MODP8192 - 112	Private Key - CSP	KAS-FFC (IKEv2)		KAS-FFC (IKEv2)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		to 200 bits				
IPsec/IKEv2 DH Public Key	Used to derive IPsec/IKEv2 DH Shared Secret	MODP-2048, MODP-3072, MODP6144, MODP8192 - 112 to 200 bits	Public Key - PSP		KAS-FFC (IKEv2)	
IPsec/IKEv2 Peer DH Public Key	Used to derive IPsec/IKEv2 DH Shared Secret	MODP-2048, MODP-3072, MODP6144, MODP8192 - 112 to 200 bits	Public Key - PSP			KAS-FFC (IKEv2)
IPsec/IKEv2 DH Shared Secret	Used to derive IPsec/IKEv2 Session Encryption Keys, IPsec/IKEv2 Authentication Keys	MODP-2048, MODP-3072, MODP6144, MODP8192 - 112 to 200 bits	Shared Secret - CSP		KAS-FFC (IKEv2)	IPsec/IKEv2 Keying Materials Development
IPsec/IKEv2 ECDH Private Key	Used to derive IPsec/IKEv2 ECDH Shared Secrets	Curve: P-384 - 192 bits	Private Key - CSP	KAS-ECC (IKEv2)		KAS-ECC (IKEv2)
IPsec/IKEv2 ECDH Public Key	Used to derive IPsec/IKEv2 ECDH Shared Secrets	Curve: P-384 - 192 bits	Public Key - PSP		KAS-ECC (IKEv2)	
IPsec/IKEv2 Peer ECDH Public Key	Used to derive IPsec/IKEv2 ECDH	Curve: P-384 - 192 bits	Public Key - PSP			KAS-ECC (IKEv2)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	Shared Secrets					
IPsec/IKEv2 ECDH Shared Secret	Used to derive IPsec/IKEv2 Session Encryption Keys, IPsec/IKEv2 Authentication Keys	Curve: P-384 - 192 bits	Shared Secret - CSP		KAS-ECC (IKEv2)	IPsec/IKEv2 Keying Materials Development
IPsec/IKEv2 RSA Private Key	Imported by the controller and used by the module to sign data for IPsec/IKEv2	Modulus 3072 bits - 128 bits	Private Key - CSP			RSA SigGen (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2)
IPsec/IKEv2 RSA Public Key	Imported by the controller used for IPsec/IKEv2 authentication	Modulus 3072 bits - 128 bits	Public Key - PSP			
IPsec/IKEv2 RSA Peer Public Key	Used for peer authentication during the IPsec/IKEv2 service	Modulus 3072 bits - 128 bits	Public Key - PSP			RSA SigVer (SSHv2, TLSv1.2, TLSv1.3, and IPsec/IKEv2)
IPsec/IKEv2 ECDSA Private Key	Imported by the controller and used by the module to sign data for IPsec/IKEv2	Curve: P-384 - 192 bits	Private Key - CSP			ECDSA SigGen (SSHv2 and IPsec/IKEv2)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
IPsec/IKEv2 ECDSA Public Key	Imported by the controller and used by IPsec/IKEv2 peer to authenticate the module	Curve: P-384 - 192 bits	Public Key - PSP			
IPsec/IKEv2 ECDSA Peer Public Key	Used for peer authentication during the IPsec/IKEv2 service	Curve: P-384 - 192 bits	Public Key - PSP			ECDSA SigVer (SSHv2 and IPsec/IKEv2)
IPsec/IKEv2 Pre-shared Secret	Used for IPsec/IKEv2 mutual authentication	16-32 bytes characters - 16-32 bytes characters	shared secret - CSP			
SKEYSEED	Keying material used to derive the IPsec/IKEv2 Session Encryption Key and IPsec/IKEv2 Authentication Key	160 bits - 160 bits	Keying Material - CSP		IPsec/IKEv2 Keying Materials Development	IPsec/IKEv2 Session Encrypt/Decrypt IPsec/IKEv2 Session Authentication
IPsec/IKEv2 Session Encryption Key	Used to secure IPsec/IKEv2 session confidentiality	128 to 256 bits - 128 to 256 bits	Session Key - CSP		IPsec/IKEv2 Keying Materials Development	IPsec/IKEv2 Session Encrypt/Decrypt
IPsec/IKEv2 Authentication Key	Used to secure IPsec/IKEv2 session integrity	at least 160 bits - at least 160 bits	Session Key - CSP		IPsec/IKEv2 Keying Materials Development	IPsec/IKEv2 Session Authentication

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SNMPv3 Shared Secret	Used for SNMPv3 user authentication	8-32 characters - N/A	Authentication Secret - CSP			
SNMPv3 Encryption Key	Used to protect SNMPv3 traffic confidentiality	128 bits - 128 bits	Encryption Key - CSP		SNMPv3 Keying Materials Development	SNMPv3 Session Encrypt/Decrypt
SNMPv3 Authentication Key	Used to secure SNMPv3 traffic integrity	At least 112 bits - At least 112 bits	Authentication Key - CSP		SNMPv3 Keying Materials Development	SNMPv3 Session Authentication
802.11i Pre-Shared Secret	Used to authenticate the User (Wireless Client)	8-32 characters - N/A	Authentication Secret - CSP			
802.11i Pairwise Master Key (PMK)	Used to derive the 802.11i Temporal Key	256 bits - 256 bits	Keying Material - CSP		802.11i Keying Materials Development	802.11i Keying Materials Development
802.11i Pairwise Transient Key (PTK)	Used to derive the 802.11i Temporal Key	384 bits - 384 bits	Keying Material - CSP		802.11i Keying Materials Development	802.11i Keying Materials Development
802.11i Temporal Key (TK)	Used to protect the 802.11i session traffic	128 bits - 128 bits	Encryption Key - CSP		802.11i Keying Materials Development	802.11i Session Encrypt/ Decrypt
802.11i Group Master Key (GMK)	Used to derive 802.11i Group Transient Key	256 bits - 256 bits	Keying Material - CSP	DRBG Function		802.11i Keying Materials Development
802.11i Group Temporal Key (GTK)	Used to protect the 802.11i group traffic	128 or 256 bits - 128 or 256 bits	Encryption Key - CSP		802.11i Keying Materials Development	802.11i Session Encrypt/ Decrypt

Table 21: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DRBG Entropy Input		DRAM:Plaintext	Until Reboot	Zeroization Method Reboot	DRBG Seed:Used With DRBG Internal State V value:Used With DRBG Key:Used With
DRBG Seed		DRAM:Plaintext	Until Reboot	Zeroization Method Reboot	DRBG Entropy Input:Used With DRBG Internal State V value:Used With DRBG Key:Used With
DRBG Internal State V value		DRAM:Plaintext	Until Reboot	Zeroization Method Reboot	DRBG Entropy Input:Used With DRBG Seed:Used With DRBG Key:Used With
DRBG Key		DRAM:Plaintext	Until Reboot	Zeroization Method Reboot	DRBG Entropy Input:Used With DRBG Seed:Used With DRBG Internal State V value:Used With
Crypto Officer Password	Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Encrypted		Zeroization Method	
User (Wireless Client) Password	Secret Input via SSHv2 encrypted by AES and HMAC	Flash:Plaintext		Zeroization Method	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	Secret Input via SSHv2 encrypted by GCM				
Factory Device Certificate		Flash:Encrypted		Zeroization Method	
Firmware Load Test Key		Flash:Plaintext		N/A	
SSHv2 ECDH Private Key		DRAM:Plaintext	While SSHv2 session is active	Zeroization Method Session Termination Reboot	SSHv2 ECDH Public Key:Paired With SSHv2 Peer ECDH Public Key:Used With
SSHv2 ECDH Public Key	Module Public Key Output	DRAM:Plaintext	While SSHv2 session is active	Zeroization Method Session Termination Reboot	SSHv2 ECDH Private Key:Paired With
SSHv2 Peer ECDH Public Key	Peer Public Key Input	DRAM:Plaintext	While SSHv2 session is active	Zeroization Method Session Termination Reboot	SSHv2 ECDH Private Key:Used With
SSHv2 ECDH Shared Secret		DRAM:Plaintext	While SSHv2 session is active	Zeroization Method Session Termination Reboot	SSHv2 ECDH Private Key:Derived From SSHv2 ECDH Public Key:Derived From
SSHv2 RSA Private Key	Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypte	Flash:Plaintext		Zeroization Method	SSHv2 RSA Public Key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroizatio n	Related SSPs
	d by GCM				
SSHv2 RSA Public Key	Module Public Key Output Secret Input via SSHv2 encrypte d by AES and HMAC Secret Input via SSHv2 encrypte d by GCM	Flash:Plaintext		Zeroization Method	SSHv2 RSA Private Key:Paired With
SSHv2 RSA Peer Public Key	Peer Public Key Input	DRAM:Plaintex t	While SSHv2 session is active	Zeroization Method Session Terminatio n Reboot	
SSHv2 ECDSA Private Key	Secret Input via SSHv2 encrypte d by AES and HMAC Secret Input via SSHv2 encrypte d by GCM	Flash:Plaintext		Zeroization Method	SSHv2 ECDSA Public Key:Paired With
SSHv2 ECDSA Public Key	Module Public Key Output Secret Input via SSHv2 encrypte d by AES and	Flash:Plaintext		Zeroization Method	SSHv2 ECDSA Private Key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	HMAC Secret Input via SSHv2 encrypted by GCM				
SSHv2 ECDSA Peer Public Key	Peer Public Key Input	DRAM:Plaintext	While SSHv2 session is active	Zeroization Method Session Termination Reboot	
SSHv2 Session Encryption Key		DRAM:Plaintext	While SSHv2 session is active	Zeroization Method Session Termination Reboot	SSHv2 Session Authentication Key:Used With
SSHv2 Session Authentication Key		DRAM:Plaintext	While SSHv2 session is active	Zeroization Method Session Termination Reboot	SSHv2 Session Encryption Key:Used With
TLSv1.2 ECDH Private Key		DRAM:Plaintext	While TLSv1.2 session is active	Zeroization Method Session Termination Reboot	TLSv1.2 ECDH Public Key:Paired With TLSv1.2 Peer ECDH Public Key:Used With
TLSv1.2 ECDH Public Key	Module Public Key Output	DRAM:Plaintext	While TLSv1.2 session is active	Zeroization Method Session Termination Reboot	TLSv1.2 ECDH Private Key:Paired With
TLSv1.2 Peer ECDH Public Key	Peer Public Key Input	DRAM:Plaintext	while TLSv1.2 session is active	Zeroization Method Session Termination Reboot	TLSv1.2 ECDH Private Key:Used With
TLSv1.2 ECDH Shared Secret		DRAM:Plaintext	While TLSv1.2 session is active	Zeroization Method Session Terminatio	TLSv1.2 ECDH Private Key:Derived From TLSv1.2 Peer

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				n Reboot	ECDH Public Key:Derived From
Locally generated TLSv1.2 RSA Private Key		Flash:Plaintext		Zeroization Method	Locally generated TLSv1.2 RSA Peer Public Key:Paired With
Locally generated TLSv1.2 RSA Public Key	Module Public Key Output	Flash:Plaintext		Zeroization Method	Locally generated TLSv1.2 RSA Private Key:Paired With
TLSv1.2 RSA Private Key	Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	TLSv1.2 RSA Public Key:Paired With
TLSv1.2 RSA Public Key	Module Public Key Output Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	TLSv1.2 RSA Private Key:Paired With
TLSv1.2 RSA Peer Public Key	Peer Public Key Input	DRAM:Plaintext	While TLSv1.2 session is active	Zeroization Method Session Termination Reboot	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLSv1.2 ECDSA Private Key	Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	TLSv1.2 ECDSA Public Key:Paired With
TLSv1.2 ECDSA Public Key	Module Public Key Output Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	TLSv1.2 ECDSA Private Key:Paired With
TLSv1.2 ECDSA Peer Public Key	Peer Public Key Input	DRAM:Plaintext	While TLSv1.2 session is active	Zeroization Method Session Termination Reboot	
TLSv1.2 Master Secret		DRAM:Plaintext	While TLSv1.2 session is active	Zeroization Method Session Termination Reboot	TLSv1.2 ECDH Shared Secret:Derived From
TLSv1.2 Session Encryption Key		DRAM:Plaintext	While TLSv1.2 session is active	Zeroization Method Session Termination Reboot	TLSv1.2 Master Secret:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLSv1.3 ECDH Private Key		DRAM:Plaintext	While TLSv1.3 session is active	Zeroization Method Session Termination Reboot	TLSv1.3 ECDH Public Key:Paired With TLSv1.3 Peer ECDH Public Key:Used With
TLSv1.3 ECDH Public Key	Module Public Key Output	DRAM:Plaintext	While TLSv1.3 session is active	Zeroization Method Session Termination Reboot	TLSv1.3 ECDH Private Key:Paired With
TLSv1.3 Peer ECDH Public Key	Peer Public Key Input	DRAM:Plaintext	While TLSv1.3 session is active	Zeroization Method Session Termination Reboot	TLSv1.3 ECDH Private Key:Used With
TLSv1.3 ECDH Shared Secret		DRAM:Plaintext	While TLSv1.3 session is active	Zeroization Method Session Termination Reboot	TLSv1.3 ECDH Private Key:Derived From TLSv1.3 Peer ECDH Public Key:Derived From
Locally generated TLSv1.3 RSA Private Key		Flash:Plaintext		Zeroization Method	Locally generated TLSv1.3 RSA Peer Public Key:Paired With
Locally generated TLSv1.3 RSA Public Key	Module Public Key Output	Flash:Plaintext		Zeroization Method	Locally generated TLSv1.3 RSA Private Key:Paired With
TLSv1.3 RSA Private Key	Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	TLSv1.3 RSA Public Key:Paired With
TLSv1.3 RSA Public Key	Module Public Key	Flash:Plaintext		Zeroization Method	TLSv1.3 RSA Private Key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	Output Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM				
TLSv1.3 RSA Peer Public Key	Peer Public Key Input	DRAM:Plaintext	While TLSv1.3 session is active	Zeroization Method Session Termination Reboot	
TLSv1.3 ECDSA Private Key	Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	TLSv1.3 ECDSA Public Key:Paired With
TLSv1.3 ECDSA Public Key	Module Public Key Output Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypte	Flash:Plaintext		Zeroization Method	TLSv1.3 ECDSA Private Key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	d by GCM				
TLSv1.3 ECDSA Peer Public Key	Peer Public Key Input	DRAM:Plaintext	While TLSv1.3 session is active	Zeroization Method Session Termination Reboot	
TLSv1.3 Master Secret		DRAM:Plaintext	While TLSv1.3 session is active	Zeroization Method Session Termination Reboot	TLSv1.3 ECDH Shared Secret:Derived From
TLSv1.3 Session Encryption Key		DRAM:Plaintext	While TLSv1.3 session is active	Zeroization Method Session Termination Reboot	TLSv1.3 Master Secret:Derived From
IPsec/IKEv2 DH Private Key		DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Termination Reboot	IPsec/IKEv2 DH Public Key:Paired With IPsec/IKEv2 Peer DH Public Key:Used With
IPsec/IKEv2 DH Public Key	Module Public Key Output	DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Termination Reboot	IPsec/IKEv2 DH Private Key:Paired With
IPsec/IKEv2 Peer DH Public Key	Peer Public Key Input	DRAM:Plaintext	while IPsec/IKEv2 tunnel is active	Zeroization Method Session Termination Reboot	IPsec/IKEv2 DH Private Key:Used With
IPsec/IKEv2 DH Shared Secret		DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Termination Reboot	SKEYSEED:Used With
IPsec/IKEv2 ECDH Private Key		DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Terminatio	IPsec/IKEv2 ECDH Public Key:Paired With IPsec/IKEv2 Peer

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				n Reboot	ECDH Public Key:Used With
IPsec/IKEv2 ECDH Public Key	Module Public Key Output	DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Termination Reboot	IPsec/IKEv2 ECDH Private Key:Paired With
IPsec/IKEv2 Peer ECDH Public Key	Peer Public Key Input	DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Termination Reboot	IPsec/IKEv2 ECDH Private Key:Used With
IPsec/IKEv2 ECDH Shared Secret		DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Termination Reboot	SKEYSEED:Used With
IPsec/IKEv2 RSA Private Key	Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	IPsec/IKEv2 RSA Public Key:Paired With
IPsec/IKEv2 RSA Public Key	Module Public Key Output Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypte	Flash:Plaintext		Zeroization Method	IPsec/IKEv2 RSA Private Key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	d by GCM				
IPsec/IKEv2 RSA Peer Public Key	Peer Public Key Input	DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Termination Reboot	
IPsec/IKEv2 ECDSA Private Key	Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	IPsec/IKEv2 ECDSA Public Key:Paired With
IPsec/IKEv2 ECDSA Public Key	Module Public Key Output Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	IPsec/IKEv2 ECDSA Private Key:Paired With
IPsec/IKEv2 ECDSA Peer Public Key	Peer Public Key Input	DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Termination Reboot	
IPsec/IKEv2 Pre-shared Secret	Secret Input via SSHv2 encrypte	DRAM:Plaintext	While IPsec/IKEv2 tunnel is active	Zeroization Method Session Terminatio	SKEYSEED:Derived to

Name	Input - Output	Storage	Storage Duration	Zeroizatio n	Related SSPs
	d by AES and HMAC Secret Input via SSHv2 encrypte d by GCM			n Reboot	
SKEYSEED		DRAM:Plaintex t	While IPsec/IKEv 2 tunnel is active	Zeroization Method Session Terminatio n Reboot	IPsec/IKEv2 DH Shared Secret:Derived From IPsec/IKEv2 ECDH Shared Secret:Derived From IPsec/IKEv2 Pre-shared Secret:Derived From
IPsec/IKEv2 Session Encryption Key		DRAM:Plaintex t	While IPsec/IKEv 2 tunnel is active	Zeroization Method Session Terminatio n Reboot	IPsec/IKEv2 DH Shared Secret:Derived From IPsec/IKEv2 ECDH Shared Secret:Derived From
IPsec/IKEv2 Authenticatio n Key		DRAM:Plaintex t	While IPsec/IKEv 2 tunnel is active	Zeroization Method Session Terminatio n Reboot	IPsec/IKEv2 DH Shared Secret:Derived From IPsec/IKEv2 ECDH Shared Secret:Derived From
SNMPv3 Shared Secret	Secret Input via SSHv2 encrypte d by AES and HMAC Secret Input via SSHv2 encrypte	DRAM:Plaintex t		Zeroization Method	SNMPv3 Encryption Key:Derive To SNMPv3 Authentication Key:Derive To

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	d by GCM				
SNMPv3 Encryption Key		DRAM:Plaintext	While SNMPv3 session is active	Zeroization Method Session Termination Reboot	SNMPv3 Shared Secret:Derived From
SNMPv3 Authentication Key		DRAM:Plaintext	While SNMPv3 session is active	Zeroization Method Session Termination Reboot	SNMPv3 Shared Secret:Derived From SNMPv3 Encryption Key:Used With
802.11i Pre-Shared Secret	Secret Input via SSHv2 encrypted by AES and HMAC Secret Input via SSHv2 encrypted by GCM	Flash:Plaintext		Zeroization Method	
802.11i Pairwise Master Key (PMK)		DRAM:Plaintext	While 802.11i session is active	Zeroization Method Session Termination Reboot	802.11i Pre-Shared Secret:Derived From
802.11i Pairwise Transient Key (PTK)		DRAM:Plaintext	While 802.11i session is active	Zeroization Method Session Termination Reboot	802.11i Pairwise Master Key (PMK):Derived From
802.11i Temporal Key (TK)		DRAM:Plaintext	While 802.11i session is active	Zeroization Method Session Termination Reboot	802.11i Pairwise Transient Key (PTK):Derived From
802.11i Group Master Key (GMK)		DRAM:Plaintext	While 802.11i	Zeroization Method Session	

Name	Input - Output	Storage	Storage Duration	Zeroizatio n	Related SSPs
			session is active	Terminatio n Reboot	
802.11i Group Temporal Key (GTK)		DRAM:Plaintex t	While 802.11i session is active	Zeroization Method Session Terminatio n Reboot	802.11i Pairwise Master Key (PMK):Derived From 802.11i Group Master Key (GMK):Derived From

Table 22: 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

112-bit security keys
The module includes an implementation of DH using MODP-2048 for shared secret computation. This implementation will be non-Approved for all uses starting January 1, 2030.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
RSA SigVer (FIPS186-5) (A6768)	RSA 4096 SigVer with SHA2-384	RSA signature verification	SW/FW Integrity	Module is in normal state	RSA SigVer

Table 23: Pre-Operational Self-Tests

The Module performs the following self-tests, which include the Pre-Operational 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). The self-test success or failure results are an output of the return value of the library load API call, which is functioning as the self-test status indicator. 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 (A6768)	128 bits	KAT	CAST	Module is in normal state	Encrypt	Power Up
AES-CBC decrypt KAT (A6768)	128 bits	KAT	CAST	Module is in normal state	Decrypt	Power Up
AES-CBC encrypt KAT (A6769)	128 bits	KAT	CAST	Module is in normal state	Encrypt	Power Up
AES-CBC decrypt KAT (A6769)	128 bits	KAT	CAST	Module is in normal state	Decrypt	Power Up
AES-CCM authenticated encrypt KAT (A3235)	256 bits	KAT	CAST	Module is in normal state	Authenticated Encrypt	Power up
AES-CCM authenticated decrypt KAT (A3235)	256 bits	KAT	CAST	Module is in normal state	Authenticated Decrypt	Power up
AES-CCM authenticated encrypt KAT (A3239)	256 bits	KAT	CAST	Module is in normal state	Authenticated Encrypt	Power up
AES-CCM authenticated decrypt KAT (A3239)	256 bits	KAT	CAST	Module is in normal state	Authenticated Decrypt	Power up
AES-GCM authenticated encrypt KAT (A6768)	256 bits	KAT	CAST	Module is in normal state	Authenticated Encrypt	Power up
AES-GCM authenticated decrypt KAT (A6768)	256 bits	KAT	CAST	Module is in normal state	Authenticated Decrypt	Power up
Counter DRBG Instantiate/Generate/Reseed KAT (A6768)	AES-128	KAT	CAST	Module is in normal state	Instantiate, Generate, and Reseed KATs	Power Up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigGen (FIPS186-5) KAT (A6768)	P-256 curve with SHA2-256	KAT	CAST	Module is in normal state	ECDSA SigGen KAT	Power Up
ECDSA SigVer (FIPS186-5) KAT (A6768)	P-256 curve with SHA2-256	KAT	CAST	Module is in normal state	ECDSA SigVer KAT	Power Up
Entropy Source RCT Start-up Health Tests	Repetition Count Test (RCT)	RCT	CAST	Module is in normal state	N/A	Power up
Entropy Source APT Start-up Health Tests	Adaptive Proportion Test (APT)	APT	CAST	Module is in normal state	N/A	Power up
Entropy Source RCT Continuous Health Tests	Repetition Count Test (RCT)	RCT	CAST	Module is in normal state	N/A	Performed continuously as entropy source is active
Entropy Source APT Continuous Health Tests	Adaptive Proportion Test (APT)	APT	CAST	Module is in normal state	N/A	Performed continuously as entropy source is active
HMAC-SHA-1 KAT (A6768)	SHA-1	KAT	CAST	Module is in normal state	HMAC-SHA-1	Power Up
HMAC-SHA-1 KAT (A6769)	SHA-1	KAT	CAST	Module is in normal state	HMAC-SHA-1	Power up
HMAC-SHA2-256 KAT (A6768)	SHA2-256	KAT	CAST	Module is in normal state	HMAC-SHA2-256	Power Up
HMAC-SHA2-256 KAT (A6769)	SHA2-256	KAT	CAST	Module is in normal state	HMAC-SHA2-256	Power up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-384 KAT (A6769)	SHA2-384	KAT	CAST	Module is in normal state	HMAC-SHA2-384	Power Up
HMAC-SHA2-512 KAT (A6768)	SHA2-512	KAT	CAST	Module is in normal state	HMAC-SHA2-512	Power Up
HMAC-SHA2-512 KAT (A6769)	SHA2-512	KAT	CAST	Module is in normal state	HMAC-SHA2-512	Power up
KAS-ECC-SSC Sp800-56Ar3 KAT (A6768)	P-256 Curve	KAT	CAST	Module is in normal state	Primitive Z KAT	Power Up
KAS-FFC-SSC Sp800-56Ar3 KAT (A6768)	MODP-2048	KAT	CAST	Module is in normal state	Primitive Z KAT	Power Up
KDF IKEv2 KAT (A6768)	N/A	KAT	CAST	Module is in normal state	N/A	Power Up
KDF SNMP KAT (A6768)	N/A	KAT	CAST	Module is in normal state	N/A	Power Up
KDF SP800-108 KAT (A6768)	N/A	KAT	CAST	Module is in normal state	N/A	Power up
KDF SSH KAT (A6768)	N/A	KAT	CAST	Module is in normal state	N/A	Power Up
RSA SigGen (FIPS186-5) KAT (A8586)	RSA 3072 bit modulus with SHA2-256	KAT	CAST	Module is in normal state	RSA SigGen KAT	Power Up
RSA SigGen (FIPS186-5) KAT (A6768)	2048 bit modulus with SHA2-256	KAT	CAST	Module is in normal state	RSA SigGen KAT	Power Up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigVer (FIPS186-5) KAT (A6768)	2048 bit modulus with SHA2-256	KAT	CAST	Module is in normal state	RSA SigVer KAT	Power Up
SHA2-384 KAT (A6768)	Output size: 384 bits	KAT	CAST	Module is in normal state	N/A	Power Up
TLS v1.2 KDF RFC7627 KAT (A6768)	N/A	KAT	CAST	Module is in normal state	N/A	Power Up
TLS v1.3 KDF KAT (A6768)	N/A	KAT	CAST	Module is in normal state	N/A	Power Up
RSA KeyGen (FIPS186-5) PCT (A6768)	2048 bit Modulus	PCT	PCT	Module is in normal state	RSA	Performs all required pair-wise consistency tests on the newly generated key pairs before the first operational use.
KAS-ECC-SSC Sp800-56Ar3 PCT (A6768)	Curve P-256 with SHA2-256	PCT	PCT	Module is in normal state	N/A	Performs all required pair-wise consistency tests on the newly generated key pairs before the first operational use.
KAS-FFC-SSC Sp800-56Ar3 PCT (A6768)	MODP-2048	PCT	PCT	Module is in	N/A	Performs all

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				normal state		required pair-wise consistency tests on the newly generated key pairs before the first operational use.
Firmware Load Test	RSA 4096 SigVer with SHA2-384	Signature Verification	SW/FW Load	Module is in normal state	RSA 4096 SigVer with SHA2-384	When firmware has been uploaded to the module

Table 24: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-5) (A6768)	RSA signature verification	SW/FW Integrity	Recommend 60 Days	Reboot

Table 25: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC encrypt KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
AES-CBC decrypt KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
AES-CBC encrypt KAT (A6769)	KAT	CAST	Recommend 60 Days	Reboot
AES-CBC decrypt KAT (A6769)	KAT	CAST	Recommend 60 Days	Reboot
AES-CCM authenticated encrypt KAT (A3235)	KAT	CAST	Recommend 60 Days	Reboot
AES-CCM authenticated decrypt KAT (A3235)	KAT	CAST	Recommend 60 Days	Reboot
AES-CCM authenticated encrypt KAT (A3239)	KAT	CAST	Recommend 60 Days	Reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CCM authenticated decrypt KAT (A3239)	KAT	CAST	Recommend 60 Days	Reboot
AES-GCM authenticated encrypt KAT (A6768)	KAT	CAST	Recommended 60 Days	Reboot
AES-GCM authenticated decrypt KAT (A6768)	KAT	CAST	Recommended 60 Days	Reboot
Counter DRBG Instantiate/Generate/Reseed KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
ECDSA SigGen (FIPS186-5) KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
ECDSA SigVer (FIPS186-5) KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
Entropy Source RCT Start-up Health Tests	RCT	CAST	Recommend 60 Days	Reboot
Entropy Source APT Start-up Health Tests	APT	CAST	Recommend 60 Days	Reboot
Entropy Source RCT Continuous Health Tests	RCT	CAST	N/A	N/A
Entropy Source APT Continuous Health Tests	APT	CAST	N/A	N/A
HMAC-SHA-1 KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
HMAC-SHA-1 KAT (A6769)	KAT	CAST	Recommended 60 Days	Reboot
HMAC-SHA2-256 KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
HMAC-SHA2-256 KAT (A6769)	KAT	CAST	Recommend 60 Days	Reboot
HMAC-SHA2-384 KAT (A6769)	KAT	CAST	Recommend 60 Days	Reboot
HMAC-SHA2-512 KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
HMAC-SHA2-512 KAT (A6769)	KAT	CAST	Recommend 60 Days	Reboot
KAS-ECC-SSC Sp800-56Ar3 KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
KAS-FFC-SSC Sp800-56Ar3 KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
KDF IKEv2 KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
KDF SNMP KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
KDF SP800-108 KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
KDF SSH KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigGen (FIPS186-5) KAT (A8586)	KAT	CAST	Recommend 60 Days	Reboot
RSA SigGen (FIPS186-5) KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
RSA SigVer (FIPS186-5) KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
SHA2-384 KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
TLS v1.2 KDF RFC7627 KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
TLS v1.3 KDF KAT (A6768)	KAT	CAST	Recommend 60 Days	Reboot
RSA KeyGen (FIPS186-5) PCT (A6768)	PCT	PCT	Recommend 60 Days	Reboot
KAS-ECC-SSC Sp800-56Ar3 PCT (A6768)	PCT	PCT	Recommend 60 Days	Reboot
KAS-FFC-SSC Sp800-56Ar3 PCT (A6768)	PCT	PCT	Recommend 60 Days	Reboot
Firmware Load Test	Signature Verification	SW/FW Load	N/A	N/A

Table 26: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Hard Error State	If self-test tests fail, the module is put into the hard error state and the module automatically reboots	Pre-Operational Firmware Integrity Test AES-CBC encrypt KAT (A6768) AES-CBC decrypt KAT (A6768) AES-CBC encrypt KAT (A6769) AES-CBC decrypt KAT (A6769) AES-CCM authenticated encrypt KAT (A3235) AES-CCM authenticated decrypt KAT (A3235) AES-CCM authenticated encrypt KAT (A3239) AES-CCM authenticated decrypt KAT (A3239) AES-GCM authenticated encrypt KAT (A6768) AES-GCM authenticated	Reboot the module	System Halt

Name	Description	Conditions	Recovery Method	Indicator
		decrypt KAT (A6768) Counter DRBG Instantiate/Generate/Reseed KAT (A6768) ECDSA SigGen (FIPS186-5) KAT (A6768) ECDSA SigVer (FIPS186-5) KAT (A6768) HMAC-SHA-1 KAT (A6768) HMAC-SHA-1 KAT (A6769) HMAC-SHA2-256 KAT (A6768) HMAC-SHA2-256 KAT (A6769) HMAC-SHA2-384 KAT (A6769) HMAC-SHA2-512 KAT (A6768) HMAC-SHA2-512 KAT (A6769) KAS-ECC-SSC Sp800-56Ar3 KAT (A6768) KAS-FFC-SSC Sp800-56Ar3 KAT (A6768) KDF IKEv2 KAT (A6768) KDF SNMP KAT (A6768) KDF SP800-108 KAT (A6768) KDF SSH KAT (A6768) RSA SigGen (FIPS186-5) KAT (A8586) RSA SigGen (FIPS186-5) KAT (A6768) RSA SigVer (FIPS186-5) KAT (A6768) SHA2-384 KAT (A6768) TLS v1.2 KDF RFC7627 KAT (A6768) TLS v1.3 KDF KAT (A6768)		
Soft Error State	If any of the less severe self-tests fail, the module will be put into the soft error state where the failed self-tests are retried until passed and an error indicator provided.	Entropy Source RCT Start-up Health Tests Entropy Source APT Start-up Health Tests Entropy Source RCT Continuous Health Tests Entropy Source APT Continuous Health Tests RSA KeyGen (FIPS186-5) PCT (A6768) KAS-ECC-SSC Sp800-56Ar3	The self-test will be retried.	Log message

Name	Description	Conditions	Recovery Method	Indicator
		PCT (A6768) KAS-FFC-SSC Sp800-56Ar3 PCT (A6768) Firmware Load Test		

Table 27: 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 integrity test and the conditional CASTs. The Module will only enter into the operational state after successfully passing the pre-operational integrity test and the conditional CASTs.

10.5 Operator Initiation of 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.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module meets all the Level 2 requirements for FIPS 140-3. The Crypto Officer must configure and enforce the following initialization steps. Operating this module without maintaining the following settings will put the module into a non-compliant state.

Physical Security: The Crypto Officer must first place the Tamper Evidence Labels (TELs) to the module as show in section 7.

Module Initialization and Configuration for Approved Mode:

> The module is not initialized upon receipt and must be configured to operate in Approved Mode by explicitly enabling it. The module is intended to operate exclusively in Approved Mode. Refer to the following configuration guidance to ensure proper setup.
>
> Access to mode selection requires the command-line interface (CLI) and is limited to the Cryptographic Officer. The module’s mode status is displayed after login.
>
> When an Access Point (AP) joins a SmartZone controller, it automatically inherits the controller's current mode. If an AP operating in Approved Mode joins a controller with Approved Mode disabled, the AP’s Approved Mode is also disabled. Conversely, if a controller in Approved Mode is joined by an AP with Approved Mode disabled, the controller will enable Approved Mode on the AP. If both the AP and controller are already operating in the same mode, the AP's mode remains unchanged.
>
> Although not recommended, if needed to disable the Approved Mode, the CO must disable the Approved mode via the controller. Once this is done, the AP will be

reconfigured and reset. From then on, the AP can be configured to use both Approved and Non-Compliant algorithms.

To bring the module back to the Approved mode, the CO must enable the Approved mode via the controller. Once this is done, the AP will be reconfigured and reset. From then on, the AP can be configured to only use Approved algorithms.

Note: To maintain compliance, APs must only be connected to a SmartZone controller operating in Approved Mode.

To connect the module to a SmartZone controller, execute the following command:

set scg <ip-address-of-controller>

The module will initiate a connection to the approved mode controller. To verify the connection status, use the following command:

get scg

The module will connect to the SmartZone controller and adopt the configuration given by the controller.

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.