F5, Inc.

FIPS 140-3 Non-Proprietary Security Policy

Device Cryptographic Module

Prepared by:
atsec information security corporation
4516 Seton Center Parkway, Suite 250
Austin, TX 78759
www.atsec.com

Prepared for:
F5, Inc.
801 Fifth Ave
Seattle, WA 98104
www.f5.com

Table of Contents

List of Tables

List of Figures

Copyrights and Trademarks

F5®, BIG-IP®, TMOS® are Registered trademarks of F5, Inc.

Intel® Xeon® and Intel® Atom® processors are Registered trademarks of Intel Corporation.

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy that contains the security rules under which the Device Cryptographic Module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an Overall Security Level 2 module.

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

1.3 Additional information:

In preparing the Security Policy document, the laboratory formatted the vendor-supplied documentation for consolidation without altering the technical statements therein contained. The further refining of the Security Policy document was conducted iteratively throughout the conformance testing. The vendor reviewed the intermediate and final Security Policy and approved all of its content.

2 Cryptographic Module Specification

2.1 Description

The Device Cryptographic Module (hereafter referred to as "the module") is a smart evolution of F5's market leading Application Delivery Controller (ADC) technology. Solutions built on this platform are load balancers. They are full proxies that give visibility into, and the power to control—inspect and encrypt or decrypt—all the traffic that passes through your network.

Purpose and Use:

Underlying BIG-IP/ VIPRION hardware and software are F5's proprietary operating system, Traffic Management Operating System (TMOS), which provides unified intelligence, flexibility, and programmability. With its application control plane architecture, TMOS is a highly optimized system providing control over the acceleration, security, and availability services your applications require. TMOS establishes a virtual, unified pool of highly scalable, resilient, and reusable services that can dynamically adapt to the changing conditions in data centers and virtual and cloud infrastructures. In the following documentation TMOS and BIG-IP are interchangeably used where system and feature modules are concerned.

The Control (or Management) Plane refers to the connection from an administrator to the BIG-IP for system management. The Data Plane refers to the traffic passed between external entities and internal servers.

Module Type: Hardware

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary: The cryptographic boundary of the module is defined by the exterior surface of the appliance (red dotted line in Figure 1). The block diagram below shows the module, its interfaces with the operational environment and the delimitation of its cryptographic boundary. Figure 1 also depicts the flow of status output (SO), control input (CI), data input (DI) and data output (DO). Description of the ports and interfaces can be found in Table- Ports and Interfaces.

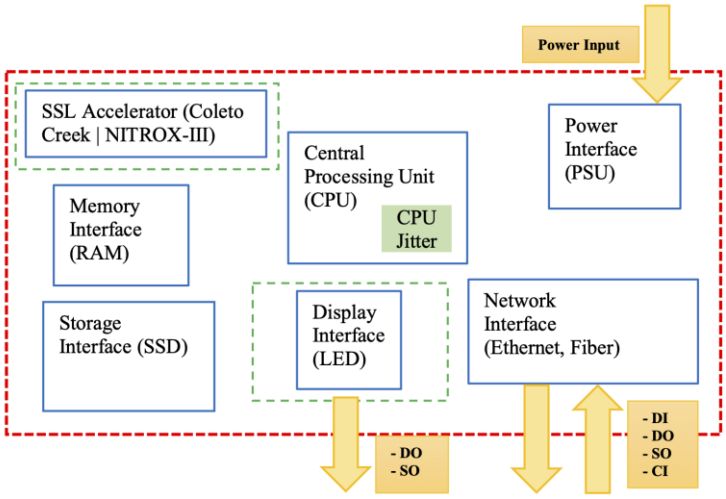

Figure 1 - Block Diagram

Diagram, Photograph:

Figure 2 – BIG-IP i15600, BIG-IP i15800, BIG-IP i15820-DF

Figure 3 - B2250 blade mounted in VIPRION chassis C2400

Figure 4 - B4450 blade mounted in VIPRION chassis C4480

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
i15600 BIG-IP iseries	i15600	17.5.0 EHF	Intel® Xeon® E5-2680v4, Broadwell	1 x USB port; 8 x 40GbE; 4 x 100GbE network ports; 1 x Console port; 1 x 1GbE management port
i15800 BIG-IP iseries	i15800	17.5.0 EHF	Intel® Xeon® E5-2680v4, Broadwell	1 x USB port; 8 x 40GbE; 4 x 100GbE network ports; 1 x Console port; 1 x 1GbE management port
i15820-DF BIG-IP iseries	i15820-DF	17.5.0 EHF	Intel® Xeon® E5-2680v4, Broadwell	1 x USB port; 8 x 40GbE; 4 x 100GbE network ports; 1 x Console port; 1 x 1GbE management port
B2250 VIPRION	C2400-B2250	17.5.0 EHF	Intel® Xeon® E5-2658v2, Ivy Bridge	2 x USB port; 4 x 40 GbE network ports; 1 x Console port; 1 x GbE management port
B4450 VIPRION	C4480-B4450	17.5.0 EHF	Intel® Xeon® E5-2658v3, Haswell	1 x USB port; 6 x 40 GbE; 2 x 100 GbE network ports; 1 x Console port; 1 x GbE (10/100/1000 Ethernet) management port

Table 2: Tested Module Identification – Hardware

2.3 Excluded Components

None

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	Automatically entered whenever an approved service is requested	Approved	Service Indicator : Approved
Non-Approved mode	Only non-approved security functions can be used	Non-Approved	Service Indicator : Not Approved

Table 3: Modes List and Description

The module enters the Approved Mode after the pre-operational self-tests and conditional algorithms self-tests (CASTs) have completed successfully.

Mode Change Instructions and Status:
The module enters the approved mode after pre-operational self-tests succeed. The module automatically switches between the approved and non-approved modes depending on the services requested by the operator. The status indicator of the mode of operation is equivalent to the indicator of the service that was requested.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A7019	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A7020	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-CCM	A7019	Key Length - 128, 192, 256	SP 800-38C
AES-CCM	A7020	Key Length - 128, 256	SP 800-38C
AES-CTR	A7019	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A7019	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
AES-GCM	A7020	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 256	SP 800-38D
Counter DRBG	A7019	Prediction Resistance - No, Yes Mode - AES-256 Derivation Function Enabled - No, Yes	SP 800-90A Rev. 1
Counter DRBG	A7020	Prediction Resistance - No Mode - AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A7019, A7020	Curve - P-256, P-384 Secret Generation Mode - testing candidates	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyVer (FIPS186-5)	A7019, A7020	Curve - P-256, P-384	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A7019, A7020	Curve - P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A7019, A7020	Curve - P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC-SHA-1	A7019	Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
HMAC-SHA2-256	A7019, A7020	Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
HMAC-SHA2-384	A7019, A7020	Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A7019, A7020	Domain Parameter Generation Methods - P-256, P-384 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A7019, A7020	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDF SSH (CVL)	A7019	Cipher - AES-128, AES-256 Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1
RSA KeyGen (FIPS186-5)	A7019	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A7019	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigGen (FIPS186-5)	A7020	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
RSA SigVer (FIPS186-5)	A7019	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A7020	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
Safe Primes Key Generation	A7019, A7020	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096	SP 800-56A Rev. 3
Safe Primes Key Verification	A7019, A7020	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
SHA2-256	A7019, A7020	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A7019, A7020	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A7019, A7020	Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1

Table 4: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
Cryptographic Key Generation (CKG)	Key Type:Asymmetric	N/A	Random bit strings required for generating the cryptographic keys is compliant with [SP 800-133Rev2] section 4 example 1

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-512	Message authentication TLS
Triple-DES, Camellia, SEED	Symmetric encryption and decryption TLS
HMAC-SHA2-256, HMAC-SHA2-512, AES-GCM	Message authentication in IPsec/ IKEv2 protocol
PKCS #1 v1.5 scheme with modulus other than 2048, 3072 or 4096 bits, for all SHA sizes; PKCS #1 v1.5 and PSS schema with modulus size 2048, 3072, 4096 bits with SHA-1, SHA2-224, SHA2-512	RSA signature generation and verification
ECDSA with curves P-256, P-384 with SHA-1, SHA2-224, SHA2-512; ECDSA using curves other than P-256 and P-384, all SHA sizes	ECDSA signature generation and verification
RSA with modulus sizes up to 16384 bits	RSA encrypt / decrypt
DSA with all key and SHA sizes	DSA domain parameter generation, domain parameter verification, key pair generation, signature generation and verification
Diffie-Hellman using MODP1024, MODP2048 groups	Shared secret computation in IPsec/IKE protocol

Name	Use and Function
MD5/ SHA-1/ SHA2-224 / SHA2-512	Key Derivation function in the context of TLS KDF
EdDSA with Ed25519	EdDSA digital signature
SHA-1, AES-ECB, RSA- signature verification	SNMP
TLS ciphersuites implemented by f5-rest-node	TLS used in SSL Orchestrator (SSLO)
RSA keypair with 2048, 3072 and 4096 (REST API)	iControl representation state transfer (REST) access
EC Diffie-Hellman Ephemeral Unified with curves other than P-256, P-384. EC Diffie-Hellman using onePassDH / StaticUnified schemes. Diffie-Hellman using groups other than ffdhe2048, ffdhe3072, ffdhe4096	Shared secret computation
Triple-DES, AES-GCM-128, AES-192, AES-256	Symmetric encryption and decryption in IPsec /IKEv2
SHA-1, SHA2-512	Message Digest

Table 6: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Key Wrapping/Unwrapping with encryption and authentication or with authenticated encryption in TLS	KTS-Wrap	Key Wrapping, Key Unwrapping in the context of TLS protocol	IG D.G:approved or allowed method from IG D.G Caveat:Key establishment methodology provides between 128 or 256 bits of key strength	AES-CBC: (A7019, A7020) HMAC-SHA2-256: (A7019, A7020) HMAC-SHA2-384: (A7019, A7020) AES-CCM: (A7019, A7020) AES-GCM: (A7019, A7020)
Key Wrapping/Unwrapping with encryption and authentication in SSH	KTS-Wrap	Key Wrapping, Key Unwrapping in the context of SSH	IG D.G:approved or allowed method from IG D.G Caveat:Key establishment methodology provides between 128 or 256 bits of key strength	AES-CBC: (A7019) AES-CTR: (A7019) HMAC-SHA2-256: (A7019) HMAC-SHA-1: (A7019)
Key pair generation	AsymKeyPair-KeyGen	Generate an ECDSA, ECDH or RSA key pair		ECDSA KeyGen (FIPS186-5): (A7019, A7020)

Name	Type	Description	Properties	Algorithms
				RSA KeyGen (FIPS186-5): (A7019)
Key pair verification	AsymKeyPair-KeyVer CKG	Verify an ECDSA or ECDH or DH key pair		ECDSA KeyVer (FIPS186-5): (A7019, A7020) Safe Primes Key Verification: (A7019, A7020) Cryptographic Key Generation (CKG): ()
Signature generation	DigSig-SigGen	Generate a digital signature		ECDSA SigGen (FIPS186-5): (A7019, A7020) RSA SigGen (FIPS186-5): (A7019, A7020)
Signature verification	DigSig-SigVer	Verify a digital signature		ECDSA SigVer (FIPS186-5): (A7019, A7020) RSA SigVer (FIPS186-5): (A7019, A7020)
Safe Primes Key Generation	AsymKeyPair-KeyGen CKG	Generate a DH key pair		Safe Primes Key Generation: (A7019, A7020) Cryptographic Key Generation (CKG): ()
Key derivation	KAS-135KDF	Key derivation using protocol KDF		TLS v1.2 KDF RFC7627: (A7019, A7020) KDF SSH: (A7019)
Random Number Generation in Control Plane	DRBG	Generate random bytes		Counter DRBG: (A7019)
Random Number Generation in Data Plane	DRBG	Generate random bytes		Counter DRBG: (A7020)
Message digest	SHA	Compute a message digest		SHA2-256: (A7019, A7020) SHA2-384: (A7019, A7020)
SSH Handshake	KAS-Full	Key agreement	IG:D.F Scenario 2 (path 2), split	KAS-ECC-SSC Sp800-56Ar3:

Name	Type	Description	Properties	Algorithms
			Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL Caveat:Key establishment methodology provides between 128 and 192-bits of security strength	(A7019) KDF SSH: (A7019)
TLS Handshake (ECC)	KAS-Full	Key agreement	IG:D.F Scenario 2 (path 2), split Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL Caveat:Key establishment methodology provides between 128 and 192-bits of security strength	KAS-ECC-SSC Sp800-56Ar3: (A7019, A7020) TLS v1.2 KDF RFC7627: (A7019, A7020)
TLS Handshake (FFC)	KAS-Full	Key agreement	IG:D.F Scenario 2 (path 2), split Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL Caveat:Key establishment methodology provides between 112 to 150-bits of security strength	KAS-FFC-SSC Sp800-56Ar3: (A7019, A7020) TLS v1.2 KDF RFC7627: (A7019, A7020)

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

AES-GCM: The IV for AES-GCM is constructed in compliance with IG C.H scenario 1a (TLS 1.2).

For TLS 1.2, the module offers the AES-GCM implementation and uses the context of Scenario 1a of IG C.H. The module is compliant with SP 800-52r2 section 3.3.1 and the mechanism for IV generation is compliant with RFC5288.

The module's implementation of AES-GCM is compliant to IG C.H option (i) where module implements TLS

protocol. The design of the TLS protocol implicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key. In the event the module's power is lost and restored, the consuming application must ensure that a new key for use with the AES-GCM key encryption or decryption under this scenario shall be established.

RSA module sizes (IG C.F): All the modulus sizes supported by the module have been ACVP tested for FIPS 186-5 RSA signature verification. The number of Miller-Rabin tests is compliant with Table B.1 of FIPS 186-5.

SP 800-56Ar3 Assurances: To comply with the assurances found in Section 5.6.2 of SP 800-56Ar3, the keys for KAS-FFC-SSC and KAS-ECC-SSC must be generated using the approved key generation services specified in section 2.9. The module performs full public key validation on the generated public keys. Additionally, the module performs full public key validation on the received public keys.

2.8 RBG and Entropy

Cert Number	Vendor Name
E74	F5

Table 8: Entropy Certificates

The entropy source used by the module specified in Table Entropy Sources uses jitter variations caused by executing instructions and memory accessed. The operator does not have the ability to modify the F5 entropy source (ES) configuration settings (see details in Public Use Document referenced in section 11.2.

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
CPU Jitter 3.4.0	Non-Physical	• BIG-IP 17.5.0 EHF on Intel Broadwell E5-2680v4 • BIG-IP 17.5.0 EHF on Intel Haswell E5-2658v3 • BIG-IP 17.5.0 EHF on Intel Ivy Bridge E5-2658v2	256 bits	Full entropy	SHA-3 vetted conditioning component. ACVP Cert. A2621

Table 9: Entropy Sources

The module employs a Deterministic Random Bit Generator (DRBG) based on [SP 800-90Ar1] for the generation of random values used in asymmetric keys, and for providing a RNG service to calling applications. The approved DRBG provided by the module is the CTR_DRBG with AES-256.

The output of entropy source provides full entropy to seed and reseed SP 800-90Ar1 DRBG during initialization (seed) and reseeding (reseed).

In accordance with FIPS 140-3 IG D.L, the 'Entropy input string', 'seed', 'DRBG internal state (V and key values)' are considered CSPs by the module.

No non-DRBG functions or instances are able to access the DRBG internal state.

2.9 Key Generation

The module implements asymmetric key generation methods according to SP 800-133r2 section 5. The key generation methods are specified in the *Security Function Implementations* table in section 2.6. The

cryptographic module performs Cryptographic Key Generation (CKG) included in table *Vendor-Affirmed Algorithms* in section 2.5.

The module does not implement symmetric key generation as an explicit service. The HMAC and AES symmetric keys are derived from shared secrets by applying [SP 800-135] as part of the TLS/ SSH protocols. The scenario maps to the [SP 800-133r2] section 6.2.1 *Symmetric keys generated using Key Agreement Scheme*.

2.10 Key Establishment

The module implements SSP agreement with key establishment as specified in the *Security Function Implementations* table in section 2.6.

The module implements SSP transport with key wrapping services specified in the *Security Function Implementations* table in section 2.6.

2.11 Industry Protocols

GCM with internal IV generation in the approved mode is compliant with version 1.2 of the TLS protocol (RFC 5288) and shall only be used in conjunction with the TLS protocol.

Additionally, the module implements the TLS 1.2 and SSH key derivation functions for use in the TLS protocol and SSH protocol (RFC 4253 and RFC 6668).

No parts of the TLS 1.2 and SSHv2 protocols, other than the KDF, have been tested by the CAVP or CMVP.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

The physical ports mapping to the logical interfaces and the flow of data passing over them are described in the Table below.

Physical Port	Logical Interface(s)	Data That Passes
Network Interface (SFP, SFP+, and QSFP+ ports (Ethernet and/or Fiber Optic) which allow transfer speeds from 1Gbps up to 100Gbps	Data Input	TLS/SSH protocol input messages; Configuration commands for interface management
Network Interface (SFP, SFP+, and QSFP+ ports)	Data Output	TLS/SSH protocol output messages; Status logs
Network Interface (SFP, SFP+, and QSFP+ ports)	Control Input	API which control system state (e.g. reset system, power-off system)
Network Interface (SFP, SFP+, and QSFP+ ports); Display Interface (LEDs, and/or output to STDOUT	Status Output	API which provides system status information
Power Interface	Power	PSU

Table 10: Ports and Interfaces

The logical interfaces are the commands through which the users of the module request services. There are no external input or output devices to the module can be used for data input, data output, status output or control input.

The module does not implement Control Output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Role-based authentication with Password (CLI or Web interface)	The password must consist of a minimum of 8 characters with at least one from each of the three-character classes. Character classes are defined as: digits (0-9), ASCII lowercase letters (a-z), ASCII uppercase letters (A-Z). Assuming a worst-case scenario where the password contains six numerical digits, one ASCII lowercase letter and one ASCII uppercase letter. The probability of guessing every character successfully is (1/10)^6 * (1/26)^1 * (1/26)^1 = 1/676,000,000. Note: this is less than 1/1,000,000. The maximum number of login attempts is limited to 3 after which the account is locked. This means that, in the worst case, an attacker has the probability of guessing the password in one minute as 3/676,000,000. Note: This is less than 1/100,000.	Password-based authentication	1/676,000,000	3/676,000,000
Role-based authentication with SSH ECDSA key-pair (CLI only)	The ECDSA using P-256 or P-384 curves for key based authentication yields a minimum security-strength of 128 bits. The chance of a random authentication attempt falsely succeeding is at most 1/(2^128) that is less than 1/1,000,000. The maximum number of login attempts is limited to 1 after which the account switches to password authentication. Then the attacker's probability to establish the connection depends on the probability of guessing the password and it is, as above, 3/676,000,000 less than 1/100,000.	ECDSA SigVer (FIPS186-5) (A7019)	1/(2^128)	3/676,000,000
Role-based authentication with SSH RSA key-pair (CLI only)	The RSA using 2048, 3092 or 4096 moduli for key based authentication yields a minimum security-strength of 112 bits. The chance of a random authentication attempt falsely succeeding is at most 1/(2^112) that is less than 1/1,000,000. The maximum number of login attempts is limited to 1 after which the account switches to password authentication. Then the attacker's	RSA SigVer (FIPS186-5) (A7019)	1/(2^112)	3/676,000,000

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	probability to establish the connection depends on the probability of guessing the password and it is, as above, 3/676,000,000 less than 1/100,000.			

Table 11: Authentication Methods

The module supports different roles (one CO role and one User role) which create different authenticated sessions, while achieving the separation between the concurrent operators. Two interfaces can be used to access the module:

- Command Line Interface (CLI): The module offers a CLI called traffic management shell (tmsh) which is accessed remotely using the SSHv2 secured session over the Ethernet connection.
- Web Interface (WebUI): The Web interface consists of HTTPS over TLS-enabled web browser which provides a graphical interface for system management tools.

The User role can access the module through CLI or WebUI. However, the CO can restrict User role access to have the User accessing through WebUI only.

The module does not maintain authenticated sessions upon power cycling. Power-cycling the system requires the authentication credentials to be re-entered. When entering password authentication data through the Web interface, any character entered will be obfuscated (i.e. replace the character entered with a dot on the entry box). When entering password authentication data through the CLI, the module does not display any character entered by the operator in stdin (e.g. keyboard).

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Administrator	Role	Crypto Officer (CO)	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
Auditor	Role	User	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
Certificate Manager	Role	User	Role-based authentication with Password (CLI or Web

Name	Type	Operator Type	Authentication Methods
			interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
Firewall Manager	Role	User	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
User Manager	Role	User	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
Manager	Role	User	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
iRule Manager	Role	User	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
Operator	Role	User	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
Resource Manager	Role	User	Role-based authentication with Password (CLI or Web interface)

Name	Type	Operator Type	Authentication Methods
			Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
Guest	Role	User	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
Log Manager	Role	User	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)
Application Editor	Role	User	Role-based authentication with Password (CLI or Web interface) Role-based authentication with SSH ECDSA key-pair (CLI only) Role-based authentication with SSH RSA key-pair (CLI only)

Table 12: Roles

At initialization of the module, the CO is the only available role. Only the CO can create the user roles.

4.3 Approved Services

The service indicator gets recorded in the high speed logging /var/log remote.log file after the service is executed. For approved services, the indicator is identified with the log message 'Service Indicator: Approved' and for non-approved services the log message includes 'Service Indicator: Not Approved'.

For SSH service the service indicator is implicit: when the SSH connection is established the service with the cipher selected is approved.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
List users	Display list of all User accounts	None	None	List of user accounts	None	Administrator User Manager

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Resource Manager Auditor
Create additional User	Create additional User	None	Username / password	Confirmation of account creation	None	Administrator - Password: W User Manager - Password: W
Modify existing Users	Modify existing Users	None	Username	Confirmation of account modification	None	Administrator User Manager
Delete User	Delete User	None	Username	Confirmation of deletion	None	Administrator User Manager
Unlock User	Remove lock from user who has exceeded login attempts	None	Username	Confirmation of unlock	None	Administrator User Manager
Update own password	Update own password	None	Own password	Confirmation of update of password	None	Administrator - Password: W Auditor - Password: W Certificate Manager - Password: W Firewall Manager - Password: W User Manager - Password: W Manager - Password:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						W iRule Manager - Password: W Operator - Password: W Resource Manager - Password: W Log Manager - Password: W Application Editor - Password: W
Update others password	Update others password	None	Username / password	Confirmation of update	None	Administrator - Password: W User Manager - Password: W
Configure Password Policy	Set password policy features	None	New password policy	Confirmation of configuration change	None	Administrator
Create TLS Certificate	Self-signed certificate creation	Service Indicator: Approved	Certificate identification information	Confirmation of certificate creation	Signature generation	Administrator - TLS RSA private key: E - TLS ECDSA private key: E Certificate Manager - TLS RSA private key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - TLS ECDSA private key: E Resource Manager - TLS RSA private key: E - TLS ECDSA private key: E
Create TLS Key	Used for the SSL Certificate key file	Service Indicator: Approved	Key identification information	Confirmation of key creation	Key pair generation Random Number Generation in Control Plane Random Number Generation in Data Plane	Administrator - TLS RSA private key: G - TLS RSA public key: G - TLS ECDSA private key: G - TLS ECDSA public key: G - DRBG seed : E - DRBG internal state (V and key values) : E,W - Entropy input: E Resource Manager - TLS RSA private key: G - TLS RSA public key: G - TLS ECDSA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						private key: G - TLS ECDSA public key: G - DRBG seed : E - DRBG internal state (V and key values) : E,W - Entropy input: E Certificate Manager - TLS RSA private key: G - TLS RSA public key: G - TLS ECDSA private key: G - TLS ECDSA public key: G - DRBG seed : E - DRBG internal state (V and key values) : E,W - Entropy input: E
Delete TLS Certificate /Key	Self-signed certificate / key deletion	None	Key identification information	Confirmation of key / certificate deletion	None	Administrator - TLS RSA private key: Z - TLS RSA public key: Z - TLS ECDSA private key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Z - TLS ECDSA public key: Z Resource Manager - TLS RSA private key: Z - TLS RSA public key: Z - TLS ECDSA private key: Z - TLS ECDSA public key: Z Certificate Manager - TLS RSA private key: Z - TLS RSA public key: Z - TLS ECDSA private key: Z - TLS ECDSA public key: Z
List Certificate	Display expiration/creation dates, certificate key size, generator of installed certificates	None	List of certificates to display	Certificate expiration information	None	Administrator Auditor Certificate Manager Resource Manager Log Manager
List Private Keys	List private key information (Name, size)	None	List of private keys to display	TLS private key information	None	Administrator Auditor Certificate

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Manager Resource Manager Log Manager
SSH establish session	SSH session key authentication, key exchanges	SSH connection successful	User / address / password / algorithms / key sizes / key derivation	Confirmation of SSH session authentication, Confirmation of SSH session key exchange	Signature verification SSH Handshake Key derivation	Administrator - SSH ECDSA public key: W - SSH RSA public key: W - Password: W - SSH EC Diffie-Hellman public key: W - SSH EC Diffie-Hellman private key: E - SSH shared secret: E,G - SSH derived session key : G Auditor - SSH ECDSA public key: W - SSH RSA public key: W - Password: W - SSH EC Diffie-Hellman public key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						W - SSH EC Diffie-Hellman private key: E - SSH shared secret: E,G - SSH derived session key : G Certificate Manager - SSH ECDSA public key: W - SSH RSA public key: W - Password: W - SSH EC Diffie-Hellman public key: W - SSH EC Diffie-Hellman private key: E - SSH shared secret: E,G - SSH derived session key : G Firewall Manager - SSH ECDSA public key: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH RSA public key: W - Password: W - SSH EC Diffie-Hellman public key: W - SSH EC Diffie-Hellman private key: E - SSH shared secret: E,G - SSH derived session key : G iRule Manager - SSH ECDSA public key: W - SSH RSA public key: W - Password: W - SSH EC Diffie-Hellman public key: W - SSH EC Diffie-Hellman private key: E - SSH shared secret: E,G - SSH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						derived session key : G Operator - SSH ECDSA public key: W - SSH RSA public key: W - Password: W - SSH EC Diffie-Hellman public key: W - SSH EC Diffie-Hellman private key: E - SSH shared secret: E,G - SSH derived session key : G Resource Manager - SSH ECDSA public key: W - SSH RSA public key: W - Password: W - SSH EC Diffie-Hellman public key: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH EC Diffie-Hellman private key: E - SSH shared secret: E,G - SSH derived session key : G User Manager - SSH ECDSA public key: W - SSH RSA public key: W - Password: W - SSH EC Diffie-Hellman public key: W - SSH EC Diffie-Hellman private key: E - SSH shared secret: E,G - SSH derived session key : G
Maintain SSH Session	SSH data encryption, decryption, integrity	SSH connection successful	SSH Derived Session key	SSH session information	Key Wrapping/Unwrapping with encryption and authentication in SSH	Administrator - SSH derived session key : E Auditor

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH derived session key : E Certificate Manager - SSH derived session key : E Firewall Manager - SSH derived session key : E iRule Manager - SSH derived session key : E Operator - SSH derived session key : E Resource Manager - SSH derived session key : E User Manager - SSH derived session key : E
Establish TLS Session	TLS session signature generation and verification, key exchange	Service Indicator: Approved	Address / algorithms/ keys / primary secret	Confirmation of establishment of the TLS session	Signature verification Message digest TLS Handshake (ECC)	Administrator - TLS RSA public key: R - TLS ECDSA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					TLS Handshake (FFC) Key derivation	public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E Auditor - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E Certificate Manager - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E Firewall Manager - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E iRule Manager

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E Operator - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E Resource Manager - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key :

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E User Manager - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						private key: E Guest - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E Manager - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E Log Manager - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W - TLS Diffie-Hellman private key: E Application Editor - TLS RSA public key: R - TLS ECDSA public key: R - TLS EC Diffie-Hellman private key: E - TLS EC Diffie-Hellman public key: W - TLS pre-primary secret : E,G - TLS derived session key : G - TLS primary secret: G - TLS Diffie-Hellman public key: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS Diffie-Hellman private key: E
Maintain TLS Session	TLS data encryption, authentication	Service Indicator: Approved	TLS Derived Session key	TLS session information	Key Wrapping/Unwrapping with encryption and authentication or with authenticated encryption in TLS	Administrator - TLS derived session key : E Auditor - TLS derived session key : E Certificate Manager - TLS derived session key : E Firewall Manager - TLS derived session key : E iRule Manager - TLS derived session key : E Operator - TLS derived session key : E Resource Manager - TLS derived session key : E User

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Manager - TLS derived session key : E Manager - TLS derived session key : E Guest - TLS derived session key : E Log Manager - TLS derived session key : E Application Editor - TLS derived session key : E
Delete ssh-keyswap	Utility service delete ssh keys	None	SSH key to delete	Confirmation of SSH key deletion	None	Administrator - SSH ECDSA public key: Z - SSH ECDSA private key: Z - SSH RSA public key: Z - SSH RSA private key: Z Resource Manager - SSH ECDSA public key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH ECDSA private key: Z - SSH RSA public key: Z - SSH RSA private key: Z
Reboot System	Restart the cryptographic module	Module reboots	None	Confirmation of system reboot	None	Administrator - TLS primary secret: Z - TLS derived session key : Z
Secure Erase	Full system zeroization	Module end of life	Selection option	Confirmation of full system zeroization	None	Administrator - TLS RSA private key: Z - TLS RSA public key: Z - TLS ECDSA private key: Z - TLS ECDSA public key: Z - SSH ECDSA public key: Z - SSH RSA private key: Z - SSH RSA public key: Z
Show version	Return the HW and FW versions and the module's name	N/A	N/A	Module name and version	None	Administrator Auditor Certificate

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Manager Firewall Manager iRule Manager Operator Resource Manager User Manager Manager Guest Log Manager Application Editor
Show Status	Return the status of the selected module parameters (rule. node, user, history, log, key, or running-config global-settings all-properties)	N/A	N/A	Module parameter status	None	Administrator Auditor Certificate Manager Firewall Manager iRule Manager Operator Resource Manager User Manager Manager Guest Log Manager Application Editor
Close TLS / SSH session	Closing TLS / SSH session	N/A	N/A	Confirmation of TLS/SSH session closure	None	Administrator - TLS EC Diffie-Hellman private key: Z - TLS EC Diffie-Hellman

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						public key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS derived session key : Z - SSH shared secret: Z - SSH derived session key : Z - SSH EC Diffie-Hellman private key: Z - SSH EC Diffie-Hellman public key: Z - TLS Diffie-Hellman public key: Z - TLS Diffie-Hellman private key: Z Auditor - TLS EC Diffie-Hellman private key: Z - TLS EC Diffie-Hellman public key: Z - TLS pre-primary

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						secret : Z - TLS primary secret: Z - TLS derived session key : Z - SSH shared secret: Z - SSH derived session key : Z - SSH EC Diffie-Hellman private key: Z - SSH EC Diffie-Hellman public key: Z - TLS Diffie-Hellman public key: Z - TLS Diffie-Hellman private key: Z Certificate Manager - TLS EC Diffie-Hellman private key: Z - TLS EC Diffie-Hellman public key: Z - TLS pre-primary secret : Z - TLS

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						primary secret: Z - TLS derived session key : Z - SSH shared secret: Z - SSH derived session key : Z - SSH EC Diffie-Hellman private key: Z - SSH EC Diffie-Hellman public key: Z - TLS Diffie-Hellman public key: Z - TLS Diffie-Hellman private key: Z Firewall Manager - TLS EC Diffie-Hellman public key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS derived session key : Z - SSH shared

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						secret: Z - SSH derived session key : Z - SSH EC Diffie-Hellman private key: Z - SSH EC Diffie-Hellman public key: Z - TLS Diffie-Hellman public key: Z - TLS Diffie-Hellman private key: Z iRule Manager - TLS EC Diffie-Hellman public key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS derived session key : Z - SSH shared secret: Z - SSH derived session key : Z - SSH EC Diffie-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Hellman private key: Z - SSH EC Diffie-Hellman public key: Z - TLS Diffie-Hellman public key: Z - TLS Diffie-Hellman private key: Z Operator - TLS EC Diffie-Hellman private key: Z - TLS EC Diffie-Hellman public key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS derived session key : Z - SSH shared secret: Z - SSH derived session key : Z - SSH EC Diffie-Hellman private key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH EC Diffie-Hellman public key: Z - TLS Diffie-Hellman public key: Z - TLS Diffie-Hellman private key: Z Resource Manager - TLS EC Diffie-Hellman public key: Z - TLS EC Diffie-Hellman private key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS derived session key : Z - SSH shared secret: Z - SSH derived session key : Z - SSH EC Diffie-Hellman private key: Z - SSH EC Diffie-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Hellman public key: Z - TLS Diffie-Hellman public key: Z - TLS Diffie-Hellman private key: Z User Manager - TLS EC Diffie-Hellman private key: Z - TLS EC Diffie-Hellman public key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS derived session key : Z - SSH derived session key : Z - SSH EC Diffie-Hellman private key: Z - SSH EC Diffie-Hellman public key: Z - TLS Diffie-Hellman

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						public key: Z - TLS Diffie-Hellman private key: Z Manager - TLS EC Diffie-Hellman private key: - TLS EC Diffie-Hellman public key: - TLS pre-primary secret : - TLS primary secret: - TLS derived session key : - SSH shared secret: - SSH derived session key : - SSH EC Diffie-Hellman private key: - SSH EC Diffie-Hellman public key: - TLS Diffie-Hellman public key: - TLS Diffie-Hellman private key: Guest - TLS EC Diffie-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Hellman private key: - TLS EC Diffie-Hellman public key: - TLS pre-primary secret : - TLS primary secret: - TLS derived session key : - SSH shared secret: - SSH derived session key : - SSH EC Diffie-Hellman private key: - SSH EC Diffie-Hellman public key: - TLS Diffie-Hellman public key: - TLS Diffie-Hellman private key: Log Manager - TLS EC Diffie-Hellman private key: - TLS EC Diffie-Hellman public key: - TLS pre-primary

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						secret : - TLS primary secret: - TLS derived session key : - SSH shared secret: - SSH derived session key : - SSH EC Diffie-Hellman private key: - SSH EC Diffie-Hellman public key: - TLS Diffie-Hellman public key: - TLS Diffie-Hellman private key: Application Editor - TLS EC Diffie-Hellman private key: - TLS EC Diffie-Hellman public key: - TLS pre-primary secret : - TLS primary secret: - TLS derived session key :

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH shared secret: - SSH derived session key : - SSH EC Diffie-Hellman private key: - SSH EC Diffie-Hellman public key: - TLS Diffie-Hellman public key: - TLS Diffie-Hellman private key:
Self-tests	Execute integrity test. Execute the CASTs	Integrity test, CASTs from section 10	N/A	Pass or fail	Key pair generation Key pair verification Signature generation Signature verification Safe Primes Key Generation Key derivation Random Number Generation in Control Plane Random Number Generation in Data Plane	Administrator Auditor Certificate Manager Firewall Manager iRule Manager Operator Resource Manager User Manager Manager Guest Log Manager Application Editor
Show license	Return license indication	N/A	N/A	FIPS license information	None	Administrator Auditor Certificate Manager Firewall

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Manager iRule Manager Operator Resource Manager User Manager Manager Guest Log Manager Application Editor
Import TLS Certificate	Import TLS Certificate	None	Certificate to import	Confirmation of import of certificate	None	Administrator - TLS RSA public key: W - TLS ECDSA public key: W Certificate Manager - TLS RSA public key: W - TLS ECDSA public key: W Resource Manager - TLS RSA public key: W - TLS ECDSA public key: W
Export Certificate File	Export Certificate File	None	Certificate to export	Exported Certificate file	None	Administrator - TLS ECDSA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						public key: R - TLS RSA public key: R Certificate Manager - TLS RSA public key: R - TLS ECDSA public key: R Resource Manager - TLS RSA public key: R - TLS ECDSA public key: R
Create ssh-keyswap	Utility service create ssh keys	Service Indicator: Approved	SSH key to create	Confirmation of SSH key creation	Key pair generation	Administrator - SSH ECDSA private key: G - SSH ECDSA public key: G - SSH RSA private key: G - SSH RSA public key: G Resource Manager - SSH ECDSA private key: G - SSH ECDSA public key: G - SSH RSA private key: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH RSA public key: G
Configure Firewall	Set policy rules, and address lists for use by firewall rules	None	Policy rules, address lists	Confirmation of policy configuration	None	Administrator Firewall Manager Resource Manager
Show firewall state	Display the current system-wide state of firewall rules	None	N/A	Display the current system wide state of the firewall rules.	None	Administrator Auditor Certificate Manager Firewall Manager User Manager Manager iRule Manager Operator Resource Manager Guest Log Manager Application Editor
Configure Firewall Users	Configure Firewall Users	None	Firewall user and configuration information	Confirmation of configuration	None	Administrator Firewall Manager User Manager
Show statistics	Shows statistics of firewall rules on the BIG-IP system	None	N/A	List of statistics of firewall rules	None	Administrator Auditor Certificate Manager Firewall Manager User Manager Manager iRule

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Manager Operator Resource Manager Guest Log Manager Application Editor
View System Audit Log	Display logs/files of configuration changes	None	N/A	Display of system audit logs	None	Administrator Auditor Resource Manager Manager iRule Manager Log Manager
Export Analytics Logs System	Export Analytics Logs System	None	N/A	Display System Analytics Logs	None	Administrator User Manager
Enable/ Disable Audit	Enable/ Disable Audit	None	N/A	Confirmation of enabling or disabling of audit	None	Administrator Resource Manager
Configure Boot Options	Enable Quiet boot, Manage boot locations	None	Boot options	Confirmation of configuration of boot options	None	Administrator Resource Manager
Configure SSH access options	Enable / Disable SSH access, Configure IP address allow list	None	SSH access / IP address list	Confirmation of configuration of SSH access options	None	Administrator Resource Manager
Configure SSH user configuration	Update ssh/ authorized_keys file for user authentication	None	ssh/ authorized_keys file	Confirmation of configuration of SSH user configuration	None	Administrator - SSH ECDSA public key: W - SSH RSA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						public key: W
Modify nodes and pool members	Enable / Disable nodes and pool members	None	Which nodes and pool members to modify	Confirmation of modification of nodes and pool members	None	Administrator Manager Resource Manager
Configure nodes	Create, modify, view, delete nodes	None	List of nodes to create / modify / view / delete	Confirmation of creation / modification / display / deletion of nodes	None	Administrator Manager Resource Manager
Configure iRules	Create, modify, view, delete, iRules	None	List of iRules to create / modify/ view/ delete	Confirmation of creation / modification / display / deletion of iRules	None	Administrator iRule Manager Resource Manager Manager

Table 13: Approved Services

For the above table, the convention below applies when specifying the access permissions (types) that the service has for each SSP.

G = **Generate**: The module generates or derives the SSP.

R = **Read**: The SSP is read from the module (e.g. the SSP is output).

W = **Write**: The SSP is updated, imported, or written to the module.

E = **Execute**: The module uses the SSP in performing a cryptographic operation.

Z = **Zeroise**: The module zeroises the SSP.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Maintain TLS session	Data encryption, Data authentication	HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-512 Triple-DES, Camellia, SEED DSA with all key and SHA sizes	CO / User
SSLO Configuration and usage	Management of the module protected by iApplx authentication	TLS ciphersuites implemented by f5-rest-node	CO / User
iControl REST access	Access to the system through REST API	RSA keypair with 2048, 3072 and 4096 (REST API)	CO / User

Name	Description	Algorithms	Role
IPsec /IKEv2	Protocol configuration	HMAC-SHA2-256, HMAC-SHA2-512, AES-GCM Diffie-Hellman using MODP1024, MODP2048 groups Triple-DES, AES-GCM-128, AES-192, AES-256	CO / User
Simple network management protocol (SNMP)	Protocol configuration	SHA-1, AES-ECB, RSA- signature verification	CO / User
Establish TLS session	Signature generation and verification, Key Exchange, Message digest	PKCS #1 v1.5 scheme with modulus other than 2048, 3072 or 4096 bits, for all SHA sizes; PKCS #1 v1.5 and PSS schema with modulus size 2048, 3072, 4096 bits with SHA-1, SHA2-224, SHA2-512 ECDSA with curves P-256, P-384 with SHA-1, SHA2-224, SHA2-512; ECDSA using curves other than P-256 and P-384, all SHA sizes RSA with modulus sizes up to 16384 bits EdDSA with Ed25519 MD5/ SHA-1/ SHA2-224 / SHA2-512 EC Diffie-Hellman Ephemeral Unified with curves other than P-256, P-384. EC Diffie-Hellman using onePassDH / StaticUnified schemes. Diffie-Hellman using groups other than ffdhe2048, ffdhe3072, ffdhe4096 SHA-1, SHA2-512	CO / User

Table 14: Non-Approved Services

4.5 External Software/Firmware Loaded

None

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is using the approved integrity technique HMAC-SHA-384. The HMAC key used for integrity test is embedded in the module.

Integrity tests are performed as part of the Pre-Operational Self-Tests.

5.2 Initiate on Demand

The on demand integrity test is performed as part of the Pre-Operational Self-Tests by powering the module off and powering it on again.

6 Operational Environment

6.1 Operational Environment Type and Requirements

The module operates in a non-modifiable operational environment provided by F5 called TMOS 17.5.0 EHF. The module is a hardware validated at a Security Level 2 in Physical Security. Once the module is operational, it does not allow the loading of any additional firmware.

There are no further requirements for this security area.

Type of Operational Environment: Non-Modifiable

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Production grade enclosure (SL1)	N/A	N/A
Opaque enclosure (SL2)	N/A	N/A
Tamper Evident Labels (SL2)	Once per month	The Crypto Officer is responsible for inspecting the quality of the tamper labels on a regular basis to confirm that the module has not been tampered with. The Crypto Officer checks the quality of the tamper evident labels for any sign of removal, replacement, tearing, etc. If any label is found to be damaged or missing, a kit providing 25 tamper labels is available for purchase.

Table 15: Mechanisms and Actions Required

7.2 User Placed Tamper Seals

Number:

Hardware Appliance	# of Tamper Labels
BIG-IP i15600 BIG-IP i15800 BIG-IP i15800-DF	7
VIPRION C2400-B2250	1
VIPRION C4480-B4450	2

Placement: The pictures below show the location of all tamper evident labels for each hardware appliance. The tamper labels are delineated with red circles in the pictures below.

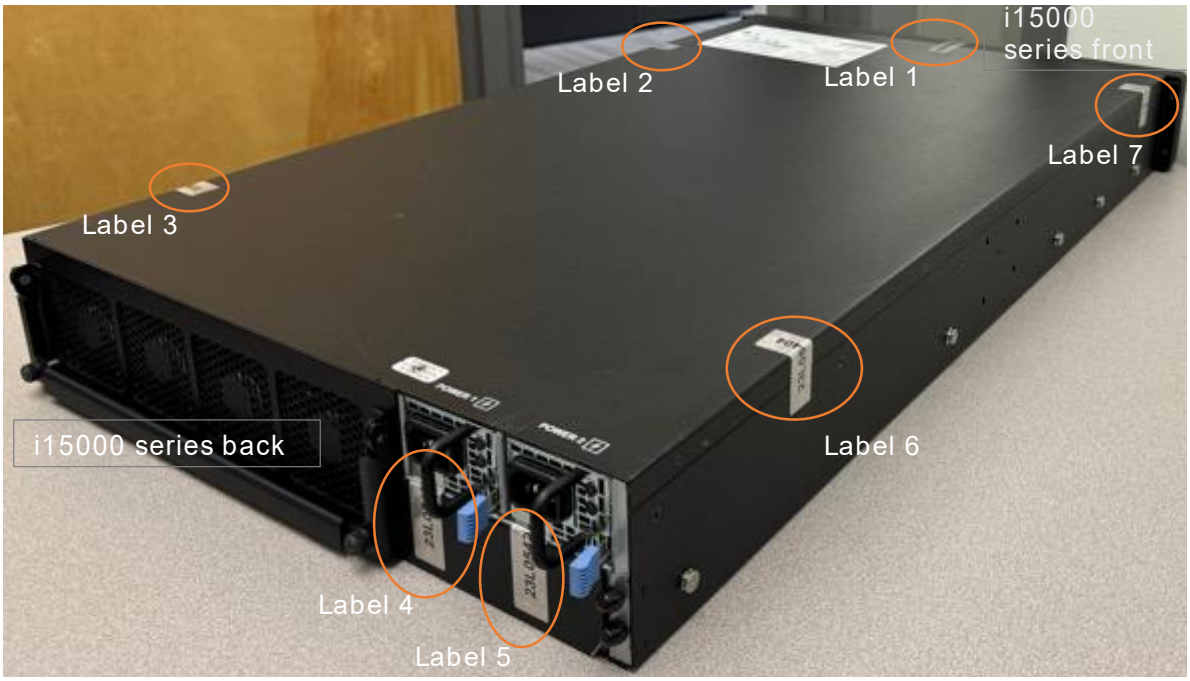

Figure 5 – Tamper labels on BIG-IP i15600, BIG-IP i15800, and BIG-IP i15820-DF.
1 label on the front, 4 labels on the sides, 2 tamper labels shown circled in orange to mark with evidence the unauthorized removal of the fan tray and PSUs (replaceable items) that give access to replaceable storage drives.

Figure 6 – Tamper labels on chassis with VIPRION B2250 blade (delineated by a red box) and three blanks (1 of 1 tamper label shown)

Figure 7 – Tamper labels on chassis with VIPRION B4450 blade
(delineated by a red box) and three blanks (2 of 2 tamper labels shown)

Surface Preparation: Before the module is installed in the production environment, tamper-evident labels must be installed in the location identified for each test platforms below. The following steps should be taken when installing or replacing the tamper evident labels on the module. The instructions are also included in *F5 Platforms: FIPS Kit Installation* provided with each module.

- Use the provided alcohol wipes to clean the chassis cover and components of dirt, grease, or oil before you apply the tamper evidence seals.
- After applying the seal, run your finger over the seal multiple times using extra high pressure.
- The seals completely cure within 24 hours.

Operator Responsible for Securing Unused Seals: The Crypto Officer shall be responsible for the storage of the label kits.

Part Numbers: P/N: F5-ADD-BIG-FIPS140

7.3 Filler Panels

Hardware Appliance	# of Filler Panels
BIG-IP i15600 BIG-IP i15800 BIG-IP i15800-DF	0
VIPRION C2400-B2250	3
VIPRION C4480-B4450	3

8 Non-Invasive Security

Per IG 12.A: Until requirements of SP 800-140F are defined, non-invasive mechanisms fall under ISO / IEC 19790:2012 Section 7.12 Mitigation of other attacks.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	The keys are stored in plaintext form and are only accessible to the authenticated operator, to which the SSPs are associated	Dynamic
SSD	The keys stored in plaintext and password will remain on the system across power cycle and are only accessible to the authenticated operator to which the SSPs/ password are associated	Static

Table 16: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Password input during SSH session	User	Module	Plaintext	Manual	Direct	
PSPs input during SSH session	User	Module	Plaintext	Automated	Electronic	Key Wrapping/Unwrapping with encryption and authentication in SSH
Public key output during TLS protocol handshake	Module	User	Plaintext	Automated	Electronic	Key Wrapping/Unwrapping with encryption and authentication or with authenticated encryption in TLS
Public key input during TLS protocol handshake	User	Module	Plaintext	Automated	Electronic	Key Wrapping/Unwrapping with encryption and authentication or with authenticated encryption in TLS
Public key input during SSH protocol handshake	User	Module	Plaintext	Automated	Electronic	Key Wrapping/Unwrapping with encryption and authentication in SSH
Public key output during SSH protocol handshake	Module	User	Plaintext	Automated	Electronic	Key Wrapping/Unwrapping with encryption and authentication in SSH

Table 17: SSP Input-Output Methods

The module only allows entry/output of public keys in plaintext from outside of the module's cryptographic boundary as part of protocol handshake process.

Once TLS/SSH session is established any key or data transfer performed thereafter is protected by authenticated encryption provided by the respective protocol.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Secure Erase	Single pass zeroization erasing the SSD contents and the module itself	All SSPs present in the module are erased including the one in the non-volatile memory	The Crypto Officer is calling the Secure Erase service which can only be triggered during reboot of the test platform
Reboot System	Clear the SSPs present in RAM memory	Volatile memory used by the module is overwritten within nanoseconds when the system is reboot.	The Crypto Officer is calling Reboot System service
Delete SSH keyswap	Destruction of the selected SSH authentication key	Zeroization service overwrites the memory occupied by keys with "zeros" or pre-defined values.	The Crypto Officer or Resource Manager are calling the Delete SSH keyswap service
Closing TLS/SSH Connection	Zeroization of all session specific keys	SSP values generated during key generation services are zeroized by the module	Closing TLS/SSH Connection

Table 18: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TLS RSA private key	RSA private key used for RSA signature generation in TLS protocol	Modulus N: 2048 and 4096-bit keys - 112 and 150-bits	Asymmetric - CSP	Key pair generation		Signature generation
TLS RSA public key	RSA public key used for RSA signature verification in TLS protocol	Modulus N: 2048 and 4096-bit keys - 112 and 150-bits	Asymmetric - PSP	Key pair generation		Signature verification
TLS ECDSA private key	ECDSA private key used for EC signature generation, shared secret computation in TLS protocol	Curve size: P-256, P-384 - 128 and 192-bits	Asymmetric - CSP	Key pair generation		Signature generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TLS ECDSA public key	ECDSA public key used for EC signature verification, shared secret computation in TLS protocol	Curve size: P-256, P-384 - 128 and 192-bits	Asymmetric - PSP	Key pair generation		Key pair verification Signature verification
TLS EC Diffie-Hellman private key	TLS EC Diffie-Hellman private key used for EC signature generation, shared secret computation in TLS protocol	Curve size: P-256, P-384 - 128 and 192-bits	Asymmetric - CSP	Key pair generation		Signature generation
TLS EC Diffie-Hellman public key	EC Diffie-Hellman public key used for signature verification, shared secret computation in TLS protocol	Curve size: P-256, P-384 - 128 and 192-bits	Asymmetric - PSP	Key pair generation		Key pair verification Signature verification
TLS pre-primary secret	TLS pre-primary secret used for deriving the TLS primary secret	Curve size: ECDH: P-256, P-384 / DH: ffdhe2048, ffdhe3072, ffdhe4096 - ECDH: 128 and 192-bits / DH: 112, 128, 150-bits	Asymmetric - CSP		TLS Handshake (ECC) TLS Handshake (FFC)	Key derivation
TLS primary secret	TLS primary secret used for deriving the TLS derived key	256-bits - 256-bits	Pre-primary secret - CSP	Key derivation		TLS Handshake (ECC)
TLS derived session key	TLS derived session key from TLS primary secret	Key length: 128 and 256-bits (AES) 112 and 256-bits (HMAC) - 128 and 256-bits (AES) 112 and 256-bits (HMAC)	Symmetric Key - CSP	Key derivation		TLS Handshake (ECC) TLS Handshake (FFC)
SSH shared secret	SSH shared secret used for deriving the SSH key	Curve size: P-256, P-384 - 128 and 192-bits	Symmetric Key - CSP		SSH Handshake	Key derivation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SSH derived session key	SSH derived session key	Key length: 128 and 256-bits (AES) 112 and 256-bits (HMAC) - 128 and 256-bits (AES) 112 and 256-bits (HMAC)	Shared secret - CSP	Key derivation		SSH Handshake
Entropy input	Compliant with IG D.L. Entropy input string used to seed the DRBG	384 bits - 384 bits	Random number generation - CSP			Random Number Generation in Control Plane Random Number Generation in Data Plane
DRBG seed	Compliant with IG D.L. DRBG seed derived from entropy input as defined in SP 800-90Ar1	384 bits - 256 bits	Random number generation - CSP	Random Number Generation in Control Plane Random Number Generation in Data Plane		Random Number Generation in Control Plane Random Number Generation in Data Plane
DRBG internal state (V and key values)	Compliant with IG D.L. Internal state of CTR_DRBG	384 bits - 256 bits	Random number generation - CSP	Random Number Generation in Control Plane Random Number Generation in Data Plane		Random Number Generation in Control Plane Random Number Generation in Data Plane
SSH ECDSA private key	ECDSA private key used for key-based authentication in SSH protocol	Curve size: P-256, P-384 - 128 and 192-bits	Asymmetric - CSP	Key pair generation		
SSH ECDSA public key	ECDSA private key used for key-based authentication in SSH protocol	Curve size: P-256, P-384 - 128 and 192-bits	Asymmetric - PSP	Key pair generation		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SSH EC Diffie-Hellman private key	EC Diffie-Hellman private key used for key exchange in SSH protocol	Curve size: P-256, P-384 - 128 and 192-bits	Asymmetric - CSP	Key pair generation		Signature generation
SSH EC Diffie-Hellman public key	EC Diffie-Hellman private key used for key exchange in SSH protocol	Curve size: P-256, P-384 - 128 and 192-bits	Asymmetric - PSP	Key pair generation		Key pair verification Signature verification
Password	Password input by the User or CO during creation of a new user or updating an existing password	8 characters - 1/676,000,000	Password - CSP			
TLS Diffie-Hellman public key	TLS Diffie-Hellman private key used for shared secret computation in TLS protocol	Curve size: ffdhe2048, ffdhe3072, ffdhe4096 - 112 to 150-bits	Asymmetric - PSP	Safe Primes Key Generation		Key pair verification Signature verification TLS Handshake (FFC)
TLS Diffie-Hellman private key	TLS Diffie-Hellman public key used for shared secret computation in TLS protocol	Curve size: ffdhe2048, ffdhe3072, ffdhe4096 - 112 to 150-bits	Asymmetric - CSP	Safe Primes Key Generation		Signature generation TLS Handshake (FFC)
SSH RSA private key	Used for key-based authentication in SSH protocol	Modulus N: 2048, 3072, 4096 bits - 112, 128, 150 bits	Asymmetric - CSP	Key pair generation		Signature generation
SSH RSA public key	Used for key-based authentication in SSH protocol	Modulus N: 2048, 3072, 4096 bits - 112, 128, 150 bits	Asymmetric - PSP	Key pair generation		Signature verification

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS RSA private key		SSD:Plaintext	From handle creation until freeing the cipher handle	Secure Erase	RSA public key:Paired With
TLS RSA public key	Public key input during TLS	SSD:Plaintext	From handle creation until	Secure Erase	RSA private key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	protocol handshake Public key output during TLS protocol handshake		freeing the cipher handle		
TLS ECDSA private key		SSD:Plaintext	From handle creation until freeing the cipher handle	Secure Erase	ECDSA public key:Paired With
TLS ECDSA public key	Public key input during TLS protocol handshake Public key output during TLS protocol handshake	SSD:Plaintext	From handle creation until freeing the cipher handle	Secure Erase	ECDSA private key:Paired With
TLS EC Diffie-Hellman private key		RAM:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	EC Diffie-Hellman public key:Paired With
TLS EC Diffie-Hellman public key	Public key input during TLS protocol handshake Public key output during TLS protocol handshake	RAM:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	EC Diffie-Hellman private key:Paired With
TLS pre-primary secret		RAM:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	TLS primary secret:Used With
TLS primary secret		RAM:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	TLS pre-primary secret :Used With
TLS derived session key		RAM:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	TLS primary secret:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SSH shared secret		RAM:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	SSH derived session key :Used With
SSH derived session key		RAM:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	SSH shared secret:Derived From
Entropy input		RAM:Plaintext	Storage duration during the usage of the CSP	Reboot System	DRBG seed :Used With
DRBG seed		RAM:Plaintext	Storage duration during the usage of the CSP	Reboot System	DRBG internal state (V and key values) :Used With
DRBG internal state (V and key values)		RAM:Plaintext	Storage duration during the usage of the CSP	Reboot System	DRBG seed :Used With
SSH ECDSA private key		SSD:Plaintext	Persistent	Secure Erase Delete SSH keyswap	
SSH ECDSA public key	PSPs input during SSH session	SSD:Plaintext	From handle creation until freeing the cipher handle	Secure Erase Delete SSH keyswap Closing TLS/SSH Connection	
SSH EC Diffie-Hellman private key		RAM:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	
SSH EC Diffie-Hellman public key	Public key output during SSH protocol handshake Public key input during SSH protocol handshake	RAM:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	
Password	Password input during SSH session	SSD:Plaintext	From handle creation until freeing the cipher handle	Closing TLS/SSH Connection	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS Diffie-Hellman public key	Public key output during TLS protocol handshake Public key input during TLS protocol handshake	SSD:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	
TLS Diffie-Hellman private key		SSD:Plaintext	From handle creation until freeing the cipher handle	Reboot System Closing TLS/SSH Connection	
SSH RSA private key		SSD:Plaintext	Persistent	Secure Erase Delete SSH keyswap	
SSH RSA public key	PSPs input during SSH session	SSD:Plaintext	From handle creation until freeing the cipher handle	Secure Erase Delete SSH keyswap Closing TLS/SSH Connection	

Table 20: SSP Table 2

9.5 Transitions

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes, starting January 1, 2031.

10 Self-Tests

At power-up the module performed the pre-operational self-tests (the integrity test) and the conditional cryptographic algorithm tests (CASTs). Both the pre-operational tests and conditional tests are performed without operator intervention, without any external controls, externally provided test vectors, output results and the determination of pass of fail is done by the module.

If the module fails any of the tests, the module transitions to the error state and a corresponding error indication is given. The module becomes inoperable, and no services are available. Data output and cryptographic operations are inhibited while the module is in the error state.

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-384 (A7019)	SHA2-384	Message Authentication	SW/FW Integrity	Module becomes operational	Integrity of the module is verified by comparing the HMAC-SHA2-384 value calculated at runtime with the HMAC-SHA2-384 value stored in the module that was computed at build time
HMAC-SHA2-384 (A7020)	SHA2-384	Message Authentication	SW/FW Integrity	Module becomes operational	Integrity of the module is verified by comparing the HMAC-SHA2-384 value calculated at runtime with the HMAC-SHA2-384 value stored in the module that was computed at build time

Table 21: Pre-Operational Self-Tests

The Pre-Operational self-tests are performed automatically when the module is powered on.

Services are not available during the pre-operational self-test and the data output interface is inhibited. On successful completion of the pre-operational self-tests, the module enters operational mode and cryptographic services are available.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Counter DRBG	AES-256 in CTR mode, with and without derivation function, prediction resistance disabled	KAT	CAST	Module becomes operational	SP 800-90ARev1 section 11.3 health tests	Test runs at power on, prior to the integrity test
AES-CBC encryption	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power on, prior

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						to the integrity test
AES-CBC decryption	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power on, prior to the integrity test
AES-GCM encryption	256-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power on, prior to the integrity test
AES-GCM decryption	256-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power on, prior to the integrity test
HMAC-SHA2-256	HMAC-SHA-256	KAT	CAST	Module becomes operational	MAC	Test runs at power on, prior to the integrity test
HMAC-SHA2-384	HMAC-SHA-384	KAT	CAST	Module becomes operational	MAC	Test runs at power on, prior to the integrity test
ECDSA KeyGen (FIPS186-5)	Requested curve size, SHA2-256	PCT	PCT	Asymmetric algorithm is performed	Calculation and verification of a digital signature	Key generation
ECDSA SigGen (FIPS186-5)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Signature generation	Test runs at power on, prior to the integrity test
ECDSA SigVer (FIPS186-5)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Signature verification	Test runs at power on, prior to the integrity test
Safe Primes Key Generation	Requested safe prime group	PCT	PCT	Asymmetric algorithm is performed	Calculation and verification of shared secret	Key generation
RSA KeyGen (FIPS186-5)	Requested modulus size, SHA2-256	PCT	PCT	Asymmetric algorithm is performed	Calculation and verification of a digital signature	Key generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigGen (FIPS186-5)	2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Signature generation	Test runs at power on, prior to the integrity test
RSA SigVer (FIPS186-5)	2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Signature verification	Test runs at power on, prior to the integrity test
KDF SSH	SHA-256	KAT	CAST	Module becomes operational	Key derivation used in the SSH protocol	Test runs at power on, prior to the integrity test
KAS-ECC-SSC Sp800-56Ar3	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power on, prior to the integrity test
KAS-FFC-SSC Sp800-56Ar3	ffdhe2048	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power on, prior to the integrity test
TLS v1.2 KDF RFC7627	SHA-256	KAT	CAST	Module becomes operational	Key derivation used in the TLS protocol	Test runs at power on, prior to the integrity test
HMAC-SHA-1	HMAC-SHA-1	KAT	CAST	Module becomes operational	MAC	Test runs at power on, prior to the integrity test
ESV - Repetition Count Test (Startup)	Startup test with 1024 samples; Cutoff value = 90	RCT	CAST	Module is operational	SP 800-90B Heath test	Performed upon startup
ESV - Repetition Count Test (Continuous)	Cutoff value = 90	RCT	CAST	Module is operational	SP 800-90B Heath test	Performed while the module is operating
ESV - Adaptive Proportional Test (Startup)	Startup test with 1024 samples; Cutoff value = 459	APT	CAST	Module is operational	SP 800-90B Heath test	Performed upon startup
ESV - Adaptive Proportional Test (Continuous)	Cutoff value = 459	APT	CAST	Module is operational	SP 800-90B Heath test	Performed while the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						module is operating

Table 22: Conditional Self-Tests

Services are not available during the Conditional self-tests and the data output interface is inhibited.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-384 (A7019)	Message Authentication	SW/FW Integrity	Determined by the operator	Module is powered-off and on
HMAC-SHA2-384 (A7020)	Message Authentication	SW/FW Integrity	Determined by the operator	Module is powered-off and on

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Counter DRBG	KAT	CAST	On Demand	Manually
AES-CBC encryption	KAT	CAST	On Demand	Manually
AES-CBC decryption	KAT	CAST	On Demand	Manually
AES-GCM encryption	KAT	CAST	On Demand	Manually
AES-GCM decryption	KAT	CAST	On Demand	Manually
HMAC-SHA2-256	KAT	CAST	On Demand	Manually
HMAC-SHA2-384	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-5)	PCT	PCT	On Demand	Manually
ECDSA SigGen (FIPS186-5)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5)	KAT	CAST	On Demand	Manually
Safe Primes Key Generation	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5)	PCT	PCT	On Demand	Manually
RSA SigGen (FIPS186-5)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-5)	KAT	CAST	On Demand	Manually
KDF SSH	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3	KAT	CAST	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627	KAT	CAST	On Demand	Manually
HMAC-SHA-1	KAT	CAST	On Demand	Manually
ESV - Repetition Count Test (Startup)	RCT	CAST	Prior to entropy generation	Automatically
ESV - Repetition Count Test (Continuous)	RCT	CAST	Prior to entropy generation	Automatically
ESV - Adaptive Proportional Test (Startup)	APT	CAST	Prior to entropy generation	Automatically
ESV - Adaptive Proportional Test (Continuous)	APT	CAST	Prior to entropy generation	Automatically

Table 24: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Halt Error	Module is no longer operational. The data output is inhibited.	Integrity test failure Failure of any of the CASTs Failure of any of the PCTs Failure of the APT, RCT at restart (power-on)	The module must be re-loaded	For integrity failure, CAST failure and health test failures at restart the module will not load. For PCT failures the module will halt and enter the error state
Health Test Error	Module is no longer operational. The data output is inhibited.	Failure of the APT, RCT at runtime	The module must be re-loaded	The module reboot in a loop

Table 25: Error States

The module must reboot to be re-loaded with a fresh image to clear the error condition.

10.5 Operator Initiation of Self-Tests

On demand and periodic self-tests are performed by powering off the module and powering it on again. This service performs the same cryptographic algorithm tests executed during pre-operational self-tests and CASTs. During the execution of the periodic and on-demand self-tests, crypto services are not available and no data output or input is possible.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

Startup Procedures: The module is distributed as a part of a BIG-IP product which includes the hardware and an installed copy of firmware with version 17.5.0 EHF. The hardware devices are shipped directly from the hardware manufacturer/authorized subcontractor via trusted carrier and tracked by that carrier. The hardware is shipped in a sealed box that includes a packing slip with a list of components inside, and with labels outside printed with the product nomenclature, sales order number, and product serial number. Upon receipt of the hardware, the customer is required to perform the following verifications:

- Ensure that the shipping label exactly identifies the correct customer name and address as well as the hardware model.
- Inspect the packaging for tampering or other issues.
- Ensure that the external labels match the expected delivery and the shipped product.
- Ensure that the components in the box match those on the documentation shipped with the product.
- Verify the hardware model with the model number given on the shipping label and marked on the hardware device itself.

Installation Process: Follow the instructions in the "*BIG-IP System: Initial Configuration*" guide for the initial setup and configuration of the module.

- Run the Setup wizard "appliance-setup-wizard" using the CLI with the CO account and default credentials. The system will prompt you to change the password.
- License the system from the WebUI. Installing the FIPS license for the host system is required for module activation. Guidance on Licensing the BIG-IP system can be found in https://support.f5.com/csp/article/K7752 and summarized as follows: Before you can activate the license for the BIG-IP system, you must obtain a base registration key. The base registration key is pre-installed on new BIG-IP systems. When you power up the product and connect to the Configuration utility, the Licensing page opens and displays the registration key. After a license activation method is selected (activation method specifies how you want the system to communicate with the F5 License Server), the F5 product generates a dossier which is an encrypted list of key characteristics used to identify the platform. If the automated activation method is selected, the BIG-IP system automatically connects to the F5 License Server and activates the license. If the manual method is selected, the Crypto Officer shall go to the F5 Product Licensing page at secure.f5.com, paste the dossier in the “Enter Your Dossier” box which produces a license. The Crypto Officer will then copy and paste it into the “License” box in the Configuration Utility. The BIG-IP system then reloads the configuration and is ready for additional system configuration.
- Once the device is installed, licensed and configured, the Crypto Officer should confirm that the system is installed and licensed correctly as follows:

  Version Confirmation: The Crypto Officer should call the show version service (with command "tmsh show sys version" and "tmsh show sys hardware"), then confirm that the provided version matches the validated version shown in Table - Tested Module Identification – Hardware. Any firmware loaded into the module other than version 17.5.0 EHF is out of the scope of this validation and will mean that the module is not operating as a FIPS validated module.

License Confirmation: The FIPS validated module activation requires installation of the license referred as 'FIPS license'. The Crypto Officer should call the show license service (with command "tmsh show sys license"), then verify that the list of license flags includes "FIPS 140-3".

Additional Guidance: The Crypto Officer should verify that the following specific configuration rules are followed in order to operate the module in the FIPS validated configuration.

- All command shells other than tmsh are not allowed. For example, bash and other user-serviceable shells are excluded.
- Management of the module via the appliance's LCD display is not allowed.
- Usage of f5-rest-node and iAppLX and provisioning of iRulesLX is not allowed.
- Only the provisioning of AFM and LTM is included.
- Remote access to the Lights Out / Always On Management capabilities of the system are not allowed.
- Serial port console and USB port should be disabled after the initial power on and communications setup of the hardware.
- Use of command run util fips-util -f init is not allowed. Running this command followed by a System Reboot service or restart will mean that the module is not operating as a FIPS validated module.
- The Single Diffie-Hellman should be turned ON for the platform GUI.
- The server ssl profile shall be configured with "cert none" and "key none" option that disables client authentication.

11.2 Administrator Guidance

The Crypto Officer should confirm that version and license are provided according to the documentation in section 11.1. The Crypto Officer should follow the additional guidance in section 11.1 to operate the module in the approved validated configuration.

The ESV Public Use Document (PUD) reference for non-physical entropy source is as follows: https://csrc.nist.gov/projects/cryptographic-module-validation-program/entropy-validations/certificate/74

11.3 Non-Administrator Guidance

N/A

11.4 Design and Rules

See section 2.7

11.5 End of Life

Secure sanitization of the module consists of using the secure erase service that will perform single pass zero write erasing the disk contents. The service can only be triggered by the administrator during reboot of the device.

12 Mitigation of Other Attacks

The module does not implement security mechanisms to mitigate other attacks.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DF	Derivation Function
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ESV	Entropy Source Validation
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
KAS	Key Agreement Schema
KAT	Known Answer Test
KW	AES Key Wrap
MAC	Message Authentication Code
NDF	No Derivation Function
NIST	National Institute of Science and Technology
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PCT	Pairwise Consistency Test
PR	Prediction Resistance
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSH	Secure Shell
TDES	Triple-DES
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

FIPS140-3	**FIPS PUB 140-3 - Security Requirements For Cryptographic Modules** March 2019 https://doi.org/10.6028/NIST.FIPS.140-3
FIPS140-3_IG	**Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program** https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
FIPS180-4	**Secure Hash Standard (SHS)** August 2015 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS186-4	**Digital Signature Standard (DSS)** July 2013 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS186-5	**Digital Signature Standard (DSS)** February 2023 https://doi.org/10.6028/NIST.FIPS.186-5
FIPS197	**Advanced Encryption Standard** November 2001 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
FIPS198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography** Specifications Version 2.1 February 2003 http://www.ietf.org/rfc/rfc3447.txt
RFC3394	**Advanced Encryption Standard (AES) Key Wrap Algorithm** September 2002 http://www.ietf.org/rfc/rfc3394.txt
RFC5649	**Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm** September 2009 http://www.ietf.org/rfc/rfc5649.txt

SP 800-38A	**NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 http://csrc.nist.gov/publications/nistpubs/800-38a/SP 800-38a.pdf
SP 800-38B	**NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication** May 2005 http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf
SP 800-38C	**NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality** May 2004 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
SP 800-38D	**NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP 800-38E	**NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices** January 2010 http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf
SP 800-38F	**NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP 800-38G	**NIST Special Publication 800-38G - Recommendation for Block Cipher Modes of Operation: Methods for Format - Preserving Encryption** March 2016 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf
SP 800-56Ar3	**NIST Special Publication 800-56A Revision 3 - Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptography** April 2018 https://doi.org/10.6028/NIST.SP.800-56Ar3
SP 800-56Cr2	**Recommendation for Key Derivation through Extraction-then-Expansion** August 2020 https://doi.org/10.6028/NIST.SP.800-56Cr2
SP 800-57	**NIST Special Publication 800-57 Part 1 Revision 4 - Recommendation for Key Management Part 1: General** January 2016 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf

SP 800-67	**NIST Special Publication 800-67 Revision 1 - Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher** January 2012 http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf
SP 800-90Ar1	**NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 http://dx.doi.org/10.6028/NIST.SP.800-90Ar1
SP 800-90B	**NIST Special Publication 800-90B - Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://doi.org/10.6028/NIST.SP.800-90B
SP 800-131r2	**Transitioning the Use of Cryptographic Algorithms and Key Lengths** March 2019 https://doi.org/10.6028/NIST.SP.800-131Ar2
SP 800-132	**NIST Special Publication 800-132 - Recommendation for Password-Based Key Derivation - Part 1: Storage Applications** December 2010 http://csrc.nist.gov/publications/nistpubs/800-132/nist-SP 800-132.pdf
SP 800-133r2	**NIST Special Publication 800-133 - Recommendation for Cryptographic Key Generation** June 2020 https://doi.org/10.6028/NIST.SP.800-133r2
SP 800-135r1	**NIST Special Publication 800-135 Revision 1 - Recommendation for Existing Application-Specific Key Derivation Functions** December 2011 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf
SP 800-140Br1	**NIST Special Publication 800-140B - CMVP Security Policy Requirements** November 2023 https://doi.org/10.6028/NIST.SP.800-140Br1