F5, Inc.

F5OS-A Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

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® are registered trademarks of F5, Inc.

Intel®, Atom® and Xeon® are registered trademarks of Intel Corporation.

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for the module with name and version defined in section 2.2. The document contains the security rules under which the 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 a 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

Purpose and Use: The F5OS-A Cryptographic Module (hereafter referred to as "the module") is a microservices-based, proprietary platform layer that provides an interface between the BIG-IP Application Delivery Controller (ADC) and the rSeries hardware.

Module Type: Firmware

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The module cryptographic boundary is defined by the 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) interfaces. Description of the ports and interfaces can be found in the *Ports and Interfaces* Table.

The Processor Algorithm Accelerators (PAA) implemented by the Intel processors listed in the Table 3 are part of the module TOEPP as shown in the Block Diagram.

The entropy source CPU Jitter is part of kernel source within the module's cryptographic boundary (see Figure 1).

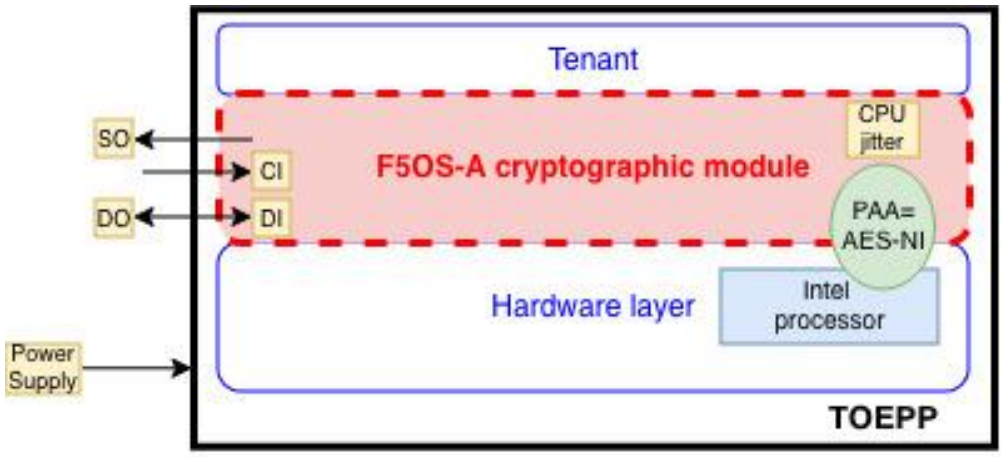

Figure 1 - Block Diagram

Tested Operational Environment's Physical Perimeter (TOEPP):

The TOEPP defined by the tested platforms listed in Table - *Tested Operational Environments - Software, Firmware, Hybrid* table is delineated by the black rectangle in Figure -Block Diagram. The TOEPP is represented by the enclosure of each of the tested platforms pictured below.

Figure 2 - r4800 isometric view

Figure 3 - r5900 front view

Figure 4 - r5920-DF front view

Figure 5 – r10900, r10920-DF front view
(same chassis for the two test platforms)

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
F5OS-A	1.5.3-40435 EHF	N/A	HMAC-SHA-384

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

The executable code is defined by the firmware version indicated in the table -Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets). All code belonging to this firmware version is the executable code of the module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
F5OS-A 1.5.3-40435 EHF	rSeries r4800	Intel® Atom® P5342 Snow Ridge NS	Yes	N/A	1.5.3-40435 EHF
F5OS-A 1.5.3-40435 EHF	rSeries r4800	Intel® Atom® P5342 Snow Ridge NS	No	N/A	1.5.3-40435 EHF
F5OS-A 1.5.3-40435 EHF	rSeries r5900	Intel® Xeon® Silver 4314 Ice Lake-SP	Yes	N/A	1.5.3-40435 EHF
F5OS-A 1.5.3-40435 EHF	rSeries r5900	Intel® Xeon® Silver 4314 Ice Lake-SP	No	N/A	1.5.3-40435 EHF
F5OS-A 1.5.3-40435 EHF	rSeries r5920-DF	Intel® Xeon® Silver 4314 Ice Lake-SP	Yes	N/A	1.5.3-40435 EHF
F5OS-A 1.5.3-40435 EHF	rSeries r5920-DF	Intel® Xeon® Silver 4314 Ice Lake-SP	No	N/A	1.5.3-40435 EHF
F5OS-A 1.5.3-40435 EHF	rSeries r10900	Intel® Xeon® Gold 6312U Ice Lake-SP	Yes	N/A	1.5.3-40435 EHF
F5OS-A 1.5.3-40435 EHF	rSeries r10900	Intel® Xeon® Gold 6312U Ice Lake-SP	No	N/A	1.5.3-40435 EHF
F5OS-A 1.5.3-40435 EHF	rSeries r10920-DF	Intel® Xeon® Gold 6312U Ice Lake-SP	Yes	N/A	1.5.3-40435 EHF

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
F5OS-A 1.5.3-40435 EHF	rSeries r10920-DF	Intel® Xeon® Gold 6312U Ice Lake-SP	No	N/A	1.5.3-40435 EHF

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

2.3 Excluded Components

The module does not claim any excluded components.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Types of Services	Status Indicator
Approved	Automatically entered whenever an approved service is requested	Approved	The service executed successfully i.e. the indicator is implicit following Example 2 of FIPS 140-3 IG 2.4.C

Table 4: Modes List and Description

Mode Change Instructions and Status:

N/A: there is no mode change.

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

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A7031, A7032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A7031, A7032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 8-128 Increment 8 Supports Counter larger than maximum value - Yes Incremental Counter - Yes Counter Tests Performed - Yes	SP 800-38A
AES-ECB	A7031, A7032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A7031, A7032	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 Payload Length - Payload Length: 128, 256, 104, 408 AAD Length - AAD Length: 128, 384, 160, 720, 0	SP 800-38D
AES-GMAC	A7031, A7032	Direction - Decrypt, Encrypt IV Generation - Internal	SP 800-38D

Algorithm	CAVP Cert	Properties	Reference
		IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 AAD Length - AAD Length: 0, 128, 160, 384, 720	
Counter DRBG	A7031	Prediction Resistance - No, Yes Supports Reseed - Yes Mode - AES-256 Derivation Function Enabled - No, Yes Additional Input - Additional Input: 0 Entropy Input - Entropy Input: 256, Entropy Input: 384 Nonce - Nonce: 0, Nonce: 128 Personalization String Length - Personalization String Length: 0-256 Increment 128, Personalization String Length: 0-384 Increment 128 Returned Bits - 512	SP 800-90A Rev. 1
Counter DRBG	A7032	Prediction Resistance - No Supports Reseed - Yes Mode - AES-256 Derivation Function Enabled - Yes Additional Input - Additional Input: 0 Entropy Input - Entropy Input: 256 Nonce - Nonce: 128 Personalization String Length - Personalization String Length: 0-256 Increment 128 Returned Bits - 512	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A7031	Curve - P-256, P-384 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A7031	Curve - P-256, P-384	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A7031	Curve - P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-4)	A7031	Component - No Curve - P-256, P-384 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA SigVer (FIPS186-5)	A7031	Component - No Curve - P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC-SHA-1	A7031, A7032	MAC - MAC: 160 Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
HMAC-SHA2-256	A7031, A7032	MAC - MAC: 256 Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
HMAC-SHA2-384	A7031, A7032	MAC - MAC: 384 Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A7031	Domain Parameter Generation Methods - P-256, P-384 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDF SSH (CVL)	A7031	Cipher - AES-128, AES-256 Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1

Algorithm	CAVP Cert	Properties	Reference
RSA KeyGen (FIPS186-5)	A7031	Key Generation Mode - probable Modulo - 2048, 3072, 4096 p mod 8 - 0 Primality Tests - 2powSecStr q mod 8 - 0 Fixed Public Exponent - 010001 Info Generated By Server - No Private Key Format - standard Public Exponent Mode - fixed	FIPS 186-5
RSA SigGen (FIPS186-5)	A7031	Hash Pair - Hash Algorithm - SHA2-256 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-4)	A7031	Signature Type - PKCS 1.5 Modulo - 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA-1 Salt Length - 20 Public Exponent Mode - Fixed Fixed Public Exponent - 010001	FIPS 186-4
RSA SigVer (FIPS186-5)	A7031	Hash Pair - Hash Algorithm - SHA2-256 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5 Fixed Public Exponent - 010001 Public Exponent Mode - fixed	FIPS 186-5
SHA-1	A7031, A7032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A7031, A7032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A7031, A7032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A7031	Hash Algorithm - SHA2-256, SHA2-384 Key Block Length - Key Block Length: 1024	SP 800-135 Rev. 1

Table 5: 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 section 4 example 1 of SP 800-133r2

Table 6: 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
AES-ECB	BC-UnAuth	Encryption/Decryption		AES-ECB: (A7031, A7032)
AES-GMAC	MAC	Message authentication code (MAC)		AES-GMAC: (A7031, A7032)
Random Number Generation	DRBG	Generate random bytes		Counter DRBG: (A7031, A7032)
Signature Generation	DigSig-SigGen	Generate a digital signature		RSA SigGen (FIPS186-5): (A7031) ECDSA SigGen (FIPS186-5): (A7031)
Signature Verification (Legacy)	DigSig-SigVer	Verify a digital signature	Compliance:FIPS 140-3 IG C.M Legacy Algorithms Hash:SHA-1 Publication:FIPS 186-4	RSA SigVer (FIPS186-4): (A7031) ECDSA SigVer (FIPS186-4): (A7031)
Signature Verification	DigSig-SigVer	Verify a digital signature	Hash:SHA2-256, SHA2-384 Publication:FIPS 186-5	RSA SigVer (FIPS186-5): (A7031) ECDSA SigVer (FIPS186-5): (A7031)
Key pair generation	AsymKeyPair-KeyGen CKG	Generate an ECDSA, ECDH or RSA key pair		RSA KeyGen (FIPS186-5): (A7031) ECDSA KeyGen (FIPS186-5): (A7031)
Key pair verification	AsymKeyPair-KeyVer	Verify an ECDSA or ECDH key pair		ECDSA KeyVer (FIPS186-5): (A7031)
SHA	SHA	Message Digest		SHA-1: (A7031, A7032) SHA2-256: (A7031,

Name	Type	Description	Properties	Algorithms
				A7032) SHA2-384: (A7031, A7032)
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	Standard:FIPS 197, SP 800-38F Compliance:approved or allowed method from IG D.G Caveat:Key establishment methodology provides between 128 or 256 bits of key strength	AES-CBC: (A7031, A7032) HMAC-SHA2-256: (A7031, A7032) HMAC-SHA2-384: (A7031, A7032) AES-GCM: (A7031, A7032)
TLS Handshake	KAS-Full	Key agreement	Compliance:IG D.F Scenario 2 (path 2) Caveat:Key establishment methodology provides between 128 or 192 bits of key strength Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL	KAS-ECC-SSC Sp800-56Ar3: (A7031) TLS v1.2 KDF RFC7627: (A7031)
SSH Handshake	KAS-Full	Key agreement	Compliance:IG D.F Scenario 2 (path 2) Caveat:Key establishment methodology provides between 128 or 192 bits of key strength Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL	KAS-ECC-SSC Sp800-56Ar3: (A7031) KDF SSH: (A7031)
Key Wrapping/Unwrapping with encryption and authentication in SSH	KTS-Wrap	Key Wrapping, Key Unwrapping in the context of SSH	Standard:FIPS 197, SP 800-38F Compliance:approved or allowed method from IG D.G Caveat:Key establishment methodology provides between 128 or 256 bits of key strength	AES-CTR: (A7031, A7032) AES-CBC: (A7031, A7032) HMAC-SHA-1: (A7031, A7032) HMAC-SHA2-256: (A7031, A7032)

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

AES GCM IV: The module implements the entire TLS protocol. AES-GCM IV is constructed in accordance with SP 800-38D in compliance with IG C.H scenario 1a. The implementation of the nonce_explicit management logic inside the module ensures that when the IV exhausts the maximum number of possible values for a given session key, the module triggers a new handshake request to establish a new key. In case the module's power is lost and then restored, the key used for the AES GCM encryption or decryption is re-distributed. The AES GCM IV generation follows [RFC 5288] and is only used for the TLS protocol version 1.2 to be compliant with [FIPS140-3_IG] IG C.H scenario 1a; thus, the module is compliant with [SP 800-52Rev2] section 3.3.1.

RSA module sizes (IG C.F): In compliance with FIPS 186-5, the RSA Signature Generation uses module sizes greater or equal to 2048 bits. All the modulus sizes supported by the module have been ACVP tested.

SP 800-56Ar3 Assurances: To comply with the assurances found in Section 5.6.2 of SP 800-56Ar3, the keys for KAS-ECC-SSC are 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.

Legacy use (IG C.M): Per SP 800-131r2, the SHA-1 with FIPS 186-4 RSA and ECDSA Digital Signature Verification is used in approved mode (for legacy use). "Algorithms designated as "Legacy" can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M

SHA-1: SHA-1 is only approved when used for Message Authentication HMAC, Message Digest and Digital signature verifications (legacy). Starting January 1, 2031 SHA-1 will be non-approved for all purposes.

2.8 RBG and Entropy

Cert Number	Vendor Name
E85	F5, Inc.

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
CPU Jitter RNG version 3.4.1 entropy source	Non-Physical	Intel Ice Lake 4314 on F5OS-A 1.5.3-40435 EHF; Intel Ice Lake 6312U on F5OS-A 1.5.3-40435 EHF; Intel Snow Ridge P5342 on F5OS-A 1.5.3-40435 EHF	256 bits	Full entropy	SHA3-256 (CAVP cert. #A3769)

Table 9: Entropy Sources

The module employs a Deterministic Random Bit Generator (DRBG) based on [SP 800-90ARev1] for the generation of random value used in asymmetric keys. The Approved DRBG provided by the module is the CTR_DRBG with AES-256. The module uses the SP 800-90B compliant entropy source specified in the *Entropy Sources* table to seed the DRBG.

The output of entropy sources provides 256-bits of entropy to seed and reseed SP 800-90ARev1 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.

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). The F5 entropy source is tested in the OE listed in the *Tested Operational Environments - Software, Firmware, Hybrid* table.

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.

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 800-133Rev2] section 6.2.1.

2.10 Key Establishment

The module implements SSP agreement, compliant with IG D.F scenario 2 (path 2). Additionally, the module implements SSP transport, compliant with IG D.G. The Key Establishment methods are specified in the *Security Function Implementations* table.

2.11 Industry Protocols

The module implements the SSH key derivation function for use in the SSH protocol (RFC 4253 updated by RFC 6668).

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 key derivation function for use in the TLS protocol.

No parts of the SSH, TLS, other than those mentioned above, have been tested by the CAVP and CMVP.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	TLS/SSH protocol input messages Configuration commands for interface management
N/A	Data Output	TLS/SSH protocol output messages Status log
N/A	Control Input	API which control system state (e.g., reset system, power-off system)
N/A	Status Output	API which provides system status information

Table 10: Ports and Interfaces

The logical interfaces are the commands through which users of the module request services. There are no external input or output devices to the module that can be used for data input, data output, status output or control input. The module does not implement a control output interface.

The physical ports are interpreted to be the physical ports of the hardware platform on which the module runs.

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 WebUI)	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 Signature Verification	1/(2^128)	3/676,000,000

Table 11: Authentication Methods

The module supports role-based authentication. The module supports concurrent operators belonging to different roles (one CO role and one User role) which create different authenticated sessions, while achieving the separation between the concurrent operators.

Two interfaces are used to access the module:

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

The CO role and User role can access the module through CLI or WebUI. The CO can restrict User role access to have the User accessing through Web Interface only. At initialization, the CO is the only available role and only the CO can create the User role.

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 WebUI, 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	CO	Role-based authentication with Password (CLI or WebUI) Role-based authentication with SSH ECDSA key-pair (CLI only)
Resource Admin	Role	User	Role-based authentication with Password (CLI or WebUI) Role-based authentication with SSH ECDSA key-pair (CLI only)
Operator	Role	User	Role-based authentication with Password (CLI or WebUI) Role-based authentication with SSH ECDSA key-pair (CLI only)
Tenant-console	Role	User	Role-based authentication with Password (CLI or WebUI) Role-based authentication with SSH ECDSA key-pair (CLI only)

Table 12: Roles

The CO and User roles are selected via the authentication credentials that are entered.

4.3 Approved Services

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 Resource Admin Operator
Create additional User	Create additional user	None	Username / password	Confirmation of account creation	None	Administrator - Password: W
Modify existing Users	Modify existing users	None	Username / modification (new username, role, password expiry date/tally count)	Confirmation of account modification	None	Administrator
Delete User	Delete existing user	None	Username	Confirmation of deletion	None	Administrator
Unlock User	Remove lock from user who has exceeded login attempts	None	Username	Confirmation of unlock	None	Administrator
Update own password	Update own password	None	Own password	Confirmation of update of password	None	Administrator - Password: W Resource Admin - Password: W Operator - Password: W
Update others password	Update others password	None	Username / password	Confirmation of update	None	Administrator - Password: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure Password Policy	Set password policy features	None	New password policy	Confirmation of configuration change	None	Administrator
Create TLS Certificate	Self-signed certificate creation	Successful creation of certificate	Certificate identification information	Confirmation of certificate creation	Signature Generation	Administrator - TLS RSA public key: W,E - TLS RSA private key: W,E - TLS ECDSA public key: W,E - TLS ECDSA private key: W,E Resource Admin - TLS RSA public key: W,E - TLS RSA private key: W,E - TLS ECDSA public key: W,E - TLS ECDSA private key: W,E
Create TLS Key	Create key for the SSL Certificate key file	Successful creation of key	Key identification information	Confirmation of key creation	Random Number Generation Key pair generation	Administrator - TLS RSA public key: G - TLS RSA private key: G - TLS ECDSA public key: G - TLS ECDSA private key: G Resource Admin - TLS RSA public key: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS RSA private key: G - TLS ECDSA public key: G - TLS ECDSA private key: G
Delete TLS Certificate / Key	Self-signed certificate / key deletion	None	Key identification information	Confirmation of key deletion	None	Administrator - TLS RSA public key: Z - TLS RSA private key: Z - TLS ECDSA public key: Z - TLS ECDSA private key: Z Resource Admin - TLS RSA public key: Z - TLS RSA private key: Z - TLS ECDSA public key: Z - TLS ECDSA private key: Z
List Certificate	Display / log expiration data of installed certificates	None	List of certificates to display	Certificate expiration information	None	Administrator Resource Admin
List private keys	List private keys	None	List of private keys to display	List of private keys	None	Administrator Resource Admin
View System Audit Log	Display logs/files of configuration changes	None	None	Display of system audit logs	None	Administrator Resource Admin
Export Analytics Logs System	Export analytics logs system	None	None	Exported system audit logs	None	Administrator

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Create Tenant	Create tenant deployment	None	password / tenant console role	Confirmation of the tenant-console role	None	Administrator - Password: W,E Resource Admin - Password: W,E
Tenant SSH establish connection	Connecting to tenant-console via SSH	None	F5 rSeries platform management address / tenant-console / password	Confirmation of Access to the tenant-console remotely over SSH	None	Tenant-console
Tenant SSH close connection	Closing the tenant-console SSH session	None	None	Confirmation of tenant-console SSH session closure	None	Tenant-console
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 Admin
Configure SSH user configuration	Update ssh/ authorized_keys file for user authentication	None	SSH ECDSA key pair (public)	Confirmation of configuration of SSH user configuration	None	Administrator - SSH ECDSA private key: W - SSH ECDSA public key : W
Reboot System	Restart the cryptographic module	Module reboots	None	Confirmation of system reboot	None	Administrator - TLS EC Diffie-Hellman public key: Z - TLS EC Diffie-Hellman private key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS AES-GCM IV: Z - TLS derived session key : Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH EC Diffie-Hellman public key: Z - SSH EC Diffie-Hellman private key: Z - SSH shared secret: Z - SSH derived session key : Z - Entropy input string : Z - DRBG seed : Z - DRBG internal state (V and key values): Z - Intermediate key generation value: Z Resource Admin - TLS EC Diffie-Hellman public key: Z - TLS EC Diffie-Hellman private key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS AES-GCM IV: Z - TLS derived session key : Z - SSH EC

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Diffie-Hellman public key: Z - SSH EC Diffie-Hellman private key: Z - SSH shared secret: Z - SSH derived session key : Z - Entropy input string : Z - DRBG seed : Z - DRBG internal state (V and key values): Z - Intermediate key generation value: Z
Secure Erase	Full system zeroization	Module end of life	Selection option	Confirmation of full system zeroization	None	Administrator - TLS RSA public key: Z - TLS RSA private key: Z - TLS ECDSA public key: Z - TLS ECDSA private key: Z - SSH ECDSA public key : Z - SSH ECDSA private key: Z - Password: Z Resource

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Admin - TLS RSA public key: Z - TLS RSA private key: Z - TLS ECDSA public key: Z - TLS ECDSA private key: Z - SSH ECDSA public key : Z - SSH ECDSA private key: Z - Password: Z
Establish SSH session	SSH key exchange	SSH connection successful	Password; SSH ECDSA private key	EC public key, Confirmation of SSH session establishment	Signature Generation Signature Verification Signature Verification (Legacy) Key pair generation Key pair verification SSH Handshake	Administrator - SSH EC Diffie-Hellman public key: G,R,W,E - SSH EC Diffie-Hellman private key: G,R,W,E - SSH shared secret: G - Intermediate key generation value: G,E,Z - SSH ECDSA public key : E - SSH ECDSA private key: E - Password: Resource Admin

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH EC Diffie-Hellman public key: G,R,W,E - SSH EC Diffie-Hellman private key: G,R,W,E - SSH shared secret: G - Intermediate key generation value: G,E,Z - SSH ECDSA public key : E - SSH ECDSA private key: E - Password: W,E Operator - SSH EC Diffie-Hellman public key: G,R,W,E - SSH EC Diffie-Hellman private key: G,R,W,E - SSH shared secret: G - Intermediate key generation value: G,E,Z - SSH ECDSA public key : E - SSH ECDSA private key: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- Password: W,E
Maintain SSH Session	SSH data encryption/decryption, SSH data integrity	SSH connection successful	Plaintext/Ciphertext	ciphertext/plaintext, MAC tag	Key Wrapping/Unwrapping with encryption and authentication in SSH	Administrator - SSH derived session key : E Resource Admin - SSH derived session key : E Operator - SSH derived session key : E
Close SSH Session	Close SSH session	SSH connection closed	None	Confirmation of SSH session closure	None	Administrator - SSH EC Diffie-Hellman public key: Z - SSH EC Diffie-Hellman private key: Z - SSH shared secret: Z - SSH derived session key : Z Resource Admin - SSH EC Diffie-Hellman public key: Z - SSH EC Diffie-Hellman private key: Z - SSH shared secret: Z - SSH derived session key : Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Operator - SSH EC Diffie-Hellman public key: Z - SSH EC Diffie-Hellman private key: Z - SSH shared secret: Z - SSH derived session key : Z
Establish TLS Session	Key exchange in TLS	Successful TLS connection	Ciphersuites	TLS EC Diffie-Hellman public key, Confirmation of establishment of TLS session	Signature Generation Signature Verification Signature Verification (Legacy) TLS Handshake	Administrator - TLS EC Diffie-Hellman public key: E - TLS EC Diffie-Hellman private key: E - TLS pre-primary secret : G,E - TLS primary secret: G,E - TLS derived session key : G - TLS AES-GCM IV: G - TLS RSA public key: E - TLS RSA private key: E - TLS ECDSA public key: E - TLS ECDSA private key: E Resource Admin

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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Maintain TLS Session	TLS data encryption/decryption, TLS data authentication	Successful TLS connection	Ciphersuites, plaintext/ciphertext	ciphertext/plaintext, MAC tag	Key Wrapping/Unwrapping with encryption and authentication or with authenticated encryption in TLS	Administrator - TLS AES-GCM IV: E - TLS derived session key : E Resource Admin - TLS AES-GCM IV: E - TLS derived session key : E Operator - TLS AES-GCM IV: E - TLS derived session key : E
Close TLS session	Close TLS session	TLS connection closed	None	Confirmation of TLS session closure	None	Administrator - TLS EC Diffie-Hellman public key: Z - TLS EC Diffie-Hellman private key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS AES-GCM IV: Z - TLS derived session key : Z Resource Admin - TLS EC Diffie-Hellman public key: Z - TLS EC

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Diffie-Hellman private key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS AES-GCM IV: Z - TLS derived session key : Z Operator - TLS EC Diffie-Hellman public key: Z - TLS EC Diffie-Hellman private key: Z - TLS pre-primary secret : Z - TLS primary secret: Z - TLS AES-GCM IV: Z - TLS derived session key : Z
Show version	Return the module name and version	None	None	Version information, and module name	None	Administrator Resource Admin Operator
Show license	Return license indication	None	None	FIPS license information	None	Administrator Resource Admin Operator
Show status	Return the module status	None	None	Status of the specific service passed in the show status command	None	Administrator Resource Admin Operator

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Self-test	Execute integrity test; Execute the CASTs	None	None	Pass/ fail results of self-tests	AES-ECB AES-GMAC Random Number Generation Signature Generation Signature Verification Key pair generation Key pair verification SHA TLS Handshake SSH Handshake Key Wrapping/Unwrapping with encryption and authentication in SSH	Administrator Resource Admin Operator
Show tenant	Lists tenant information	None	None	Lists tenant information	None	Administrator Resource Admin Operator

Table 13: Approved Services

For SSH and TLS services the service indicator is implicit via the successfully establishment of the SSH or TLS connections.

The following variables are used in the Access rights to keys or SSPs column:

- **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

N/A for this module.

4.5 External Software/Firmware Loaded

The section is not applicable for the module.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified 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 pre-operational self-tests, including the integrity test on demand, are performed by rebooting the module.

6 Operational Environment

EHF Operational Environment Type and Requirements

The module operates in a non-modifiable operational environment provided by F5 with a firmware version 1.5.3-40435 EHF. The module is a firmware 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

How Requirements are Satisfied: Once the module is operational, it does not allow the loading of any additional firmware.

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 Seals (SL2)	Once per month	The CO checks the quality of the tamper evident labels for any sign of removal, replacement, tearing. If the tamper evident labels require replacement, a kit providing 25 tamper labels is available for purchase (P/N: F5-ADD-BIG-FIPS140).

Table 14: Mechanisms and Actions Required

7.2 User Placed Tamper Seals

Number:

Hardware Appliance	# of Tamper Labels
r4800	5
r5900	4
r5920-DF	5
r10900 r10920-DF	5

Placement:

The pictures below show the location of all tamper evident labels for each hardware appliance listed in the Tested Operational Environments - Software, Firmware, Hybrid table. The tamper labels are delineated with red circles.

The tamper evident labels shall be installed for the module to operate in approved mode of operation

Surface Preparation: The following steps should be taken when installing or replacing the tamper evident labels on the test platform on which the module runs. The instructions are also included in *F5 Platforms: FIPS Kit Installation* provided with the hardware platform.

- 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: Crypto Officer

Part Numbers: F5-ADD-BIG-FIPS140

Figure 6 - Tamper labels on r4800 (5 of 5 tamper labels)

Figure 7 – Tamper labels on r5900 (4 of 4 tamper labels)

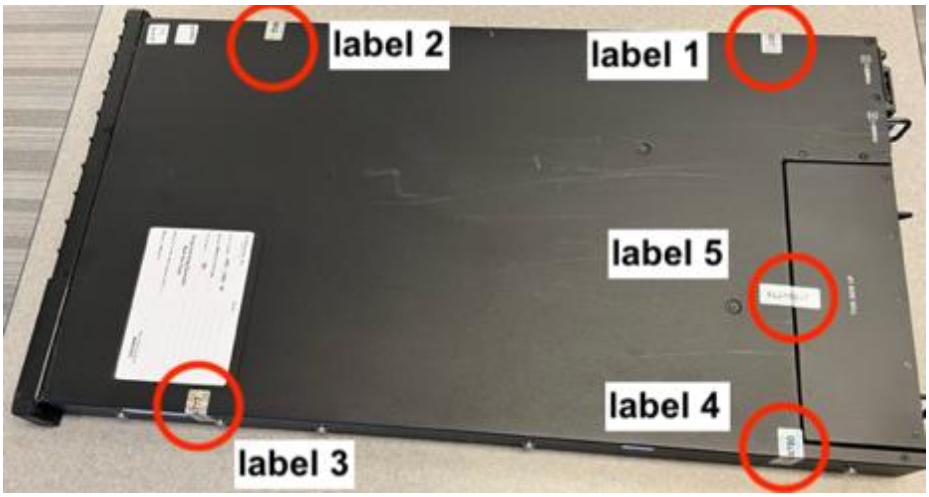

Figure 8 - Tamper labels on r5920-DF
(5 of 5 tamper labels). Labels are located on the lateral sides of the platform -labels 1,2,3 and 4. The tamper label 5 on the chassis lid is covering the ventilation fan tray that allows access to SSD.

Figure 9 – Tamper labels on r10900, r10920-DF
(4 +1 tamper labels shown). Labels are located on the lateral sides of the platform -labels 1,2,3 and 4. The tamper label 5 on the chassis lid is covering the ventilation fan tray that allows access to SSDs.

7.3 Filler Panels

Hardware Appliance	# of Filler Panels
r4800	1 (blank in Power Supply -PSU- slot)
r5900	1 (blank in PSU slot)
r5920-DF	1 (blank in PSU slot)
r10900 r10920-DF	0

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
SSD	The keys are stored in plaintext form. The static SSPs remain on the system across power cycle. SSPs are only accessible to the authenticated operator, to which the SSPs are associated.	Static
RAM	The keys are stored in plaintext form. The memory occupied by SSPs is allocated by regular memory allocation operating system calls.	Dynamic

Table 15: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Public key output during protocol handshake	Module	User	Plaintext	Automated	Electronic	Key Wrapping/Unwrapping with encryption and authentication or with authenticated encryption in TLS
Public SSPs during protocol handshake	User	Module	Plaintext	Automated	Electronic	Key Wrapping/Unwrapping with encryption and authentication in SSH
Input over encrypted TLS/SSH session	User	Module	Encrypted	Automated	Electronic	Key Wrapping/Unwrapping with encryption and authentication in SSH

Table 16: SSP Input-Output Methods

The module only allows entry/output of public keys in plaintext from outside of the module's TOEPP as part of protocol handshake process. There are no encrypted SSPs that are directly entered.

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

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Secure Erase	Single pass zeroization erasing the entire module	All SSPs present in the module are erased including the one in the non-volatile memory	The Administrator and Resource Admin calling Reboot System service 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 power is removed	The Administrator and Resource Admin calling Reboot System service
Closing TLS/SSH Connection	Zeroization of temporary values	SSP temporary values generated during key generation services are zeroized by the module	Closing TLS/SSH Connection

Table 17: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TLS RSA public key	RSA public key used for Digital signature verification in TLS protocol	Modulus N: 2048, 3072, 4096 bits - 112-150 bits	Asymmetric - PSP	Key pair generation		Signature Verification
TLS RSA private key	RSA private key pair used for digital signature generation in TLS protocol	Modulus N: 2048, 3072, 4096 bits - 112-150 bits	Asymmetric - CSP	Key pair generation		Signature Generation
TLS ECDSA public key	ECDSA public key used for digital signature verification in TLS protocol	Curve size: P-256, P-384 - 128 and 192 bits	Asymmetric - PSP	Key pair generation		Signature Verification
TLS ECDSA private key	ECDSA private key used for digital signature generation 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	ECDH public key used in TLS protocol key exchange	Curve size: P-256, P-384 - 128 and 192 bits	Asymmetric - PSP	Key pair generation		TLS Handshake
TLS EC Diffie-Hellman private key	ECDH private key used in TLS protocol key exchange	Curve size: P-256, P-384 - 128 and 192 bits	Asymmetric - CSP	Key pair generation		TLS Handshake
TLS pre-primary secret	TLS pre-primary secret used for deriving the TLS primary secret	Curve size: P-256, P-384 - 128 and 192 bits	Asymmetric - CSP		TLS Handshake	TLS Handshake
TLS primary secret	TLS primary secret derived from TLS pre-primary secret	384 bits - 128 or 192-bits	Pre-primary secret - CSP		TLS Handshake	TLS Handshake
TLS derived session key	TLS derived session key derived from TLS primary secret	Key Length: 128 and 256 bits (AES); 112-256 bits (HMAC) - 112-256 bits	Symmetric - CSP	TLS Handshake		
TLS AES-GCM IV	The AES-GCM IV is used only in the cypher suites for TLS 1.2 protocol	96 bits - 96 bits	PSP - CSP	TLS Handshake		
SSH ECDSA public key	ECDSA public key used for SSH key-based authentication	Curve size: P-256, P-384 - 128 and 192 bits	Asymmetric - PSP			Signature Verification
SSH ECDSA private key	ECDSA private key used for SSH key-based authentication	Curve size: P-256, P-384 - 128 and 192 bits	Asymmetric - CSP			Signature Generation
SSH EC Diffie-Hellman public key	EC Diffie-Hellman public key used for SSH handshake	Curve size: P-256, P-384 - 128 and 192 bits	Asymmetric - PSP	Key pair generation		SSH Handshake
SSH EC Diffie-Hellman private key	EC Diffie-Hellman private key used for SSH handshake	Curve size: P-256, P-384 - 128 and 192 bits	Asymmetric - CSP	Key pair generation		SSH Handshake
SSH shared secret	SSH shared secret established during SSH	Curve size: P-256, P-384 - 128 and 192 bits	Shared secret - CSP		SSH Handshake	SSH Handshake

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	shared secret computation					
SSH derived session key	SSH derived session key derived from SSH shared secret	Key Length: 128 and 256 bits (AES); 112-256 bits (HMAC) - 128-256 bits	Symmetric - CSP	SSH Handshake		
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			
Entropy input string	Entropy obtained from the non-physical entropy source	384 bits - 384 bits	Entropy - CSP			Random Number Generation
DRBG seed	DRBG seed derived from the entropy input string	384 bits - 256 bits	Seed - CSP	Random Number Generation		Random Number Generation
DRBG internal state (V and key values)	DRBG internal state derived from the DRBG seed	384 bits - 256 bits	Internal state - CSP	Random Number Generation		Random Number Generation
Intermediate key generation value	Temporary value generated during key pair generation services	256-4096 bits - 112-256 bits	Intermediate value - CSP	Key pair generation		Key pair generation TLS Handshake SSH Handshake

Table 18: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS RSA public key	Public key output during protocol handshake	SSD :Plaintext	From service invoked to service completed or module reboot	Secure Erase	TLS RSA private key:Paired With Intermediate key generation value:Derived From
TLS RSA private key		SSD :Plaintext	From service invoked to service completed or module reboot	Secure Erase	TLS RSA public key:Paired With Intermediate key generation value:Derived From
TLS ECDSA public key	Public key output during protocol handshake	SSD :Plaintext	From service invoked to service completed or module reboot	Secure Erase	TLS RSA private key:Paired With Intermediate key generation value:Derived From
TLS ECDSA private key		SSD :Plaintext	From service invoked to service completed or module reboot	Secure Erase	TLS ECDSA public key:Paired With Intermediate key generation value:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS EC Diffie-Hellman public key	Public key output during protocol handshake	RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	TLS EC Diffie-Hellman private key:Paired With Intermediate key generation value:Derived From
TLS EC Diffie-Hellman private key		RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	TLS EC Diffie-Hellman public key:Paired With Intermediate key generation value:Derived From
TLS pre-primary secret		RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	TLS primary secret:Derives
TLS primary secret		RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	TLS pre-primary secret :Derived From TLS derived session key :Derives
TLS derived session key	Public SSPs during protocol handshake	RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	TLS primary secret:Derived From
TLS AES-GCM IV	Public SSPs during protocol handshake	RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	TLS primary secret:Derived From
SSH ECDSA public key	Public SSPs during protocol handshake	SSD :Plaintext	From service invoked to service completed or module reboot	Secure Erase Closing TLS/SSH Connection	SSH ECDSA private key:Paired With
SSH ECDSA private key		SSD :Plaintext	From service invoked to service completed or module reboot	Secure Erase	SSH ECDSA public key :Paired With
SSH EC Diffie-Hellman public key	Public key output during protocol handshake	RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	SSH EC Diffie-Hellman private key:Paired With SSH shared secret:Establishes Intermediate key generation value:Derived From
SSH EC Diffie-Hellman private key		RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	SSH EC Diffie-Hellman public key:Paired With SSH shared secret:Establishes Intermediate key generation value:Derived From
SSH shared secret		RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	SSH EC Diffie-Hellman public key:Established By SSH EC Diffie-Hellman private key:Established By

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					SSH derived session key :Derives
SSH derived session key	Public SSPs during protocol handshake	RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System Closing TLS/SSH Connection	SSH shared secret:Derived From
Password	Input over encrypted TLS/SSH session	RAM:Plaintext	N/A	Secure Erase Closing TLS/SSH Connection	
Entropy input string		RAM:Plaintext	Storage duration during the usage of the CSP	Reboot System	DRBG seed :Derives
DRBG seed		RAM:Plaintext	Storage duration during the usage of the CSP	Reboot System	Entropy input string :Derived From DRBG internal state (V and key values):Derives
DRBG internal state (V and key values)		RAM:Plaintext	Storage duration during the usage of the CSP	Reboot System	DRBG seed :Derived From
Intermediate key generation value		RAM:Plaintext	From service invoked to service completed or module reboot	Reboot System	TLS RSA public key:Derives TLS RSA private key:Derives TLS ECDSA public key:Derives TLS ECDSA private key:Derives SSH EC Diffie-Hellman public key:Derives SSH EC Diffie-Hellman private key:Derives TLS EC Diffie-Hellman public key:Derives TLS EC Diffie-Hellman private key:Derives

Table 19: 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, 2030.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-384 (A7031)	HMAC key: 384-bits	Message authentication	SW/FW Integrity	Module is operational and services are available for use	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 (A7032)	HMAC key: 384-bits	Message authentication	SW/FW Integrity	Module is operational and services are available for use	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 20: Pre-Operational Self-Tests

At power-up the module performs the pre-operational self-tests (the integrity test) and the conditional cryptographic algorithm self-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. Services are not available during the pre-operational self-test and CASTs and the data output interface is inhibited.

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.

10.2 Conditional Self-Tests

N/A for this module.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-384 (A7031)	Message authentication	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-384 (A7032)	Message authentication	SW/FW Integrity	On Demand	Manually

Table 21: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM encrypt	Encrypt KAT	CAST	On Demand	Manually
AES-GCM decrypt	Decrypt KAT	CAST	On Demand	Manually
AES-ECB encrypt	Encrypt KAT	CAST	On Demand	Manually
AES-ECB decrypt	Decrypt KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5)	Sign KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5)	Verify KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5)	Sign KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5)	Verify KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3	Shared secret computation	CAST	On Demand	Manually
HMAC-SHA-1	HMAC KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256	HMAC KAT	CAST	On Demand	Manually
HMAC-SHA2-384	HMAC KAT	CAST	On Demand	Manually
KDF SSH	SSH KDF KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627	TLS 1.2 KDF KAT	CAST	On Demand	Manually
RSA KeyGen (FIPS186-5)	RSA PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5)	ECDH PCT	PCT	On Demand	Manually
Counter DRBG	CTR_DRBG 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 22: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error State	Module is no longer operational. The data output is inhibited.	HMAC-SHA2-384 KAT failure or HMAC-SHA2-384 integrity test failure Failure of any of the CASTs Failure of any of the PCTs Failure of the APT, RCT at restart (power-on) Failure of the APT, RCT at runtime	The module must reboot or be re-loaded with a fresh image	Module will not load after failing any of the CASTs, the integrity test or APT/RCT at restart (power on; Module will reboot after failing a PCT or APT/RCT at runtime

Table 23: Error States

10.5 Operator Initiation of Self-Tests

The software integrity tests, and cryptographic algorithm self-tests can be invoked on demand by rebooting the module. During the execution of the periodic and on-demand self-tests, crypto services are not available, and no data output or input is possible. The PCTs are executed on demand during the key generation functions invocation.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

11.1.1 Delivery and Operation

- 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's name and address as well as the hardware model.
    - 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 platform itself.

The Crypto Officer must verify that the following specific configuration rules are followed in order to operate the module in the approved mode 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/85.

The information regarding installing tamper evident labels is specified in section 7 of this document.

11.1.2 Installing F5OS

Follow the instructions in the "Initial Configuration" guide for the initial setup and configuration of the hardware module.

- Install the FIPS validated F5OS iSO onto the device. Guidance on installing or upgrading the ISO can be found here: https://techdocs.f5.com/en-us/f5os-a-1-0-0/f5-rseries-systems-installation-upgrade/title-install-upgrade-software.html#install-upgrade-options).
- 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. Guidance on Licensing the F5OS system can be found in https://techdocs.f5.com/en-us/hardware/f5-rseries-systems-getting-started/gs-system-initial-config.html#run-setup-wizard) and summarized as followed: Before you can activate the license for the F5OS system, you must obtain a base registration key. The base registration key is pre-installed on new F5OS systems. When you power up the product and connect through the WebUI, you can open the SYSTEM SETTINGS > Licensing page to display the registration key. Select "Automatic" for the license Activation Method 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 and activates the license.
- After rebooting the F5OS system, it will then be in the approved mode and is now ready for additional system configuration.
- Once the module is installed, licensed and configured, the Crypto Officer should confirm that the system is installed and licensed correctly.

11.1.3 Version Confirmation

The Crypto Officer should call the show version service (with command "show system image"), then confirm that the provided version matches the validated version shown in the *Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)* table. Any firmware loaded into the module other than version 1.5.3-40435 EHF is out of the scope of this validation and will mean that the module is not operating as a FIPS validated module.

11.1.4 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 "show system licensing"), then verify that the list of license flags includes "FIPS 140 License".

11.2 Administrator Guidance

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

- The integrity check must not be disabled. The CO can verify whether this is enabled by using the command "show system security integrity-check".
- Management of the module via the platform's LCD display is not allowed.
- Serial port console and USB port should be disabled after the initial power on and communications setup of the hardware.

The Approved mode of operation is specified in section 2.4. The administrative functions are specified in the *Approved Services* table. All the physical ports and logical interfaces are specified in section 3.1.

11.3 Non-Administrator Guidance

N/A

11.4 Design and Rules

The Crypto Officer and User shall consider the requirements and restrictions in section 2.7 when using the module.

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 and Resource Admin roles 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
CAVS	Cryptographic Algorithm Validation Scheme
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
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ESV	Entropy Source Validation
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
KWP	AES Key Wrap with Padding
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSD	Solid State Drive
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) March 2012 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
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/sp800-38a.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 800-56Br2	NIST Special Publication 800-56B Revision 2 Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography March 2019 https://doi.org/10.6028/NIST.SP.800-56Br2
SP 800-56C	NIST Special Publication 800-56C Revision 2 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 5 - Recommendation for Key Management Part 1: General May 2020 https://doi.org/10.6028/NIST.SP.800-57pt1r5
SP 800-67	NIST Special Publication 800-67 Revision 2 - Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher November 2017 https://doi.org/10.6028/NIST.SP.800-67r2

SP 800-90A	NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators June 2015 https://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 800-108	NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions (Revised) August 2022 https://doi.org/10.6028/NIST.SP.800-108r1
SP 800-131A	NIST Special Publication 800-131A Revision 1- Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths November 2015 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf
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-sp800-132.pdf
SP 800-133	NIST Special Publication 800-133 Revision 2 - Recommendation for Cryptographic Key Generation June 2020 https://doi.org/10.6028/NIST.SP.800-133r2
SP 800-135	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 800-140B	NIST Special Publication 800-140B - CMVP Security Policy Requirements March 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140B.pdf