FIPS 140-3 Non-Proprietary Security Policy

F5, Inc.

F5OS-A Cryptographic Module

Module Version: 1.7.0

FIPS Security Level 2

Document Version 1.1

Last update: 05-06-2025

Prepared by:

atsec information security corporation

4516 Seton Center Parkway, Suite 250

Austin, TX 78759

www.atsec.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 F5OS-A Cryptographic Module with firmware version 1.7.0. 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	N/A
6	Operational environment	N/A
7	Physical security	2
8	Non-invasive security	N/A
9	Sensitive security parameter management	2
10	Self-tests	2
11	Life-cycle assurance	2
12	Mitigation of other attacks	N/A
	Overall Level	2

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use: The 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 ADC and the rSeries hardware.

Module Type: Firmware

Module Embodiment: MultiChipStand

Cryptographic Boundary:

The module cryptographic boundary is defined by the red dotted line in Figure 2. The TOEPP is defined by the tested platform listed in the *Tested Operational Environments - Software, Firmware, Hybrid* table and delineated by the black rectangle in Figure 2. Figure 2 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.

Tested Operational Environment's Physical Perimeter (TOEPP):

The figure below shows the platform on which the module was tested.

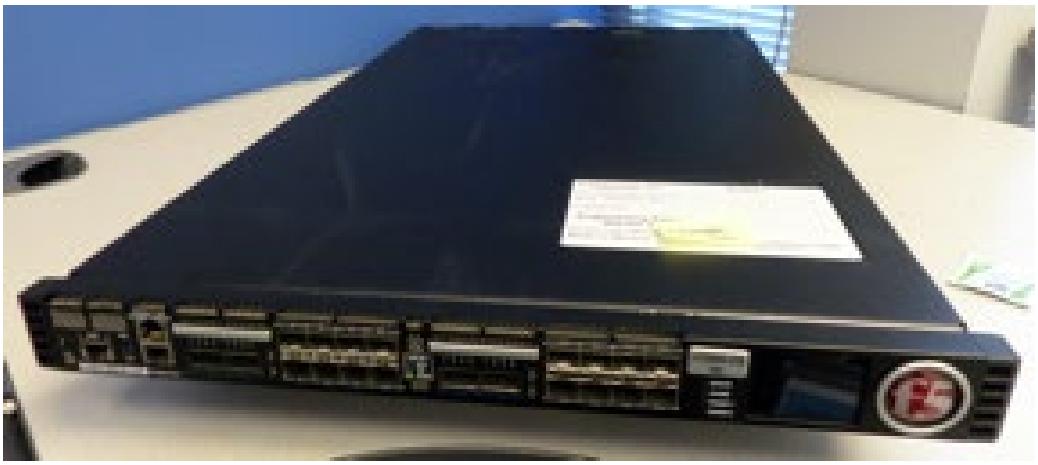

Figure 1: r12900 Platform Front View

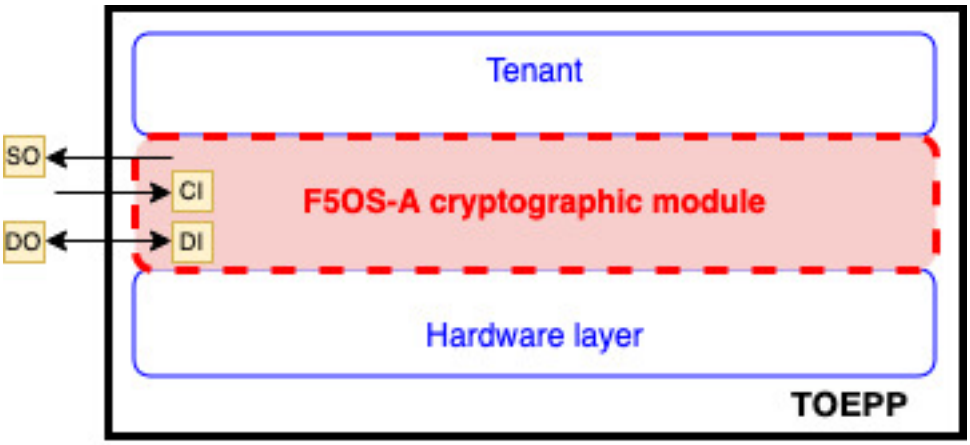

Figure 2: Block Diagram

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	1.7.0	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 1.7.0. 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 1.7.0	rSeries r12900	Intel(R) Xeon(R) Platinum 8531N Ice Lake-SP	Yes	N/A	1.7.0

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	Type	Status Indicator
Approved	Automatically entered whenever an approved service is requested	Approved	Equivalent to the indicator of the requested service as defined in section 4.3

Mode Name	Description	Type	Status Indicator
Non-Approved	Automatically entered whenever a non-approved service is requested	Non-Approved	Equivalent to the indicator of the requested service as defined in section 4.3

Table 4: Modes List and Description

The module supports two modes of operation:

- In Approved mode of operation only approved or vendor affirmed security functions can be used.
- In non-Approved mode of operation only non-approved security functions can be used.

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

Mode Change Instructions and Status:

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. SSPs used or stored in the Approved mode are not used in the non-Approved mode, and vice versa.

Degraded Mode Description:

Not applicable.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4985	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A5261	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A4985	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A5261	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4985	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A5261	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A4985	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
AES-GCM	A5261	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
AES-GMAC	A4985	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	
AES-GMAC	A5261	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
Counter DRBG	A4985	Prediction Resistance - No, Yes Mode - AES-256 Derivation Function Enabled - No, Yes	SP 800-90A Rev. 1
Counter DRBG	A5261	Prediction Resistance - No, Yes Mode - AES-256 Derivation Function Enabled - No, Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A4985	Curve - P-256, P-384 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyGen (FIPS186-5)	A5261	Curve - P-256, P-384 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A4985	Curve - P-256, P-384	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A5261	Curve - P-256, P-384	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A4985	Curve - P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384 Component - No	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A5261	Curve - P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A4985	Curve - P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A5261	Curve - P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384	FIPS 186-5
HMAC-SHA-1	A4985	Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
HMAC-SHA-1	A5261	Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
HMAC-SHA2-256	A4985	Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
HMAC-SHA2-256	A5261	Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
HMAC-SHA2-384	A4985	Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
HMAC-SHA2-384	A5261	Key Length - Key Length: 8, 16, 64, 128, 1024	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4985	Domain Parameter Generation Methods - P-256, P-384 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A5261	Domain Parameter Generation Methods - P-256, P-384 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDF SSH (CVL)	A4985	Cipher - AES-128, AES-256 Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1
KDF SSH (CVL)	A5261	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)	A4985	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA KeyGen (FIPS186-5)	A5261	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A4985	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigGen (FIPS186-5)	A5261	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A4985	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A5261	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
SHA-1	A4985	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA-1	A5261	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A4985	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A5261	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A4985	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A5261	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A4985	Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1
TLS v1.2 KDF RFC7627 (CVL)	A5261	Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1

Table 5: Approved Algorithms

Vendor-Affirmed Algorithms:

N/A for this module.

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
AES-CCM	Symmetric Encryption, Symmetric Decryption
AES-CFB	Symmetric Encryption, Symmetric Decryption
AES-OFB	Symmetric Encryption, Symmetric Decryption
AES-KW	Symmetric Encryption, Symmetric Decryption
DES	Symmetric Encryption, Symmetric Decryption
RC4	Symmetric Encryption, Symmetric Decryption
Triple-DES	Symmetric Encryption, Symmetric Decryption
SM2	Symmetric Encryption, Symmetric Decryption

Name	Use and Function
SM4	Symmetric Encryption, Symmetric Decryption
RSA Asymmetric Encryption and Decryption	Asymmetric Encryption, Asymmetric Decryption
RSA Key Generation with modulus sizes other than 2048, and 4096-bits	Key generation
DSA	Domain Parameter Generation, Domain Parameter Verification, Key Pair Generation, Digital Signature Generation, Digital Signature Verification
EdDSA Signature Generation and Verification using Ed25519	Digital Signature Generation, Digital Signature Verification
ECDSA Key Generation and Verification with curves other than P-256 and P-384	Key Generation, Key Verification
Safe Primes Key Generation and Verification for Diffie-Hellman	Key Generation, Key Verification
RSA PKCS#1 v1.5 Signature Generation and Verification using 2048, 3072 or 4096-bits modulus with SHA-1, SHA2-224, SHA2-512	Digital Signature Generation, Digital Signature Verification
RSA PKCS #1 v1.5 Signature Generation and Verification with modulus other than 2048, 3072 or 4096 bits, for all SHA sizes	Digital Signature Generation, Digital Signature Verification
RSA PSS Signature Generation and Verification using 2048, 3072 or 4096-bits modulus	Digital Signature Generation, Digital Signature Verification
ECDSA Signature Generation and Verification using curves other than P-256 and P-384, all SHA sizes	Digital Signature Generation, Digital Signature Verification
ECDSA Signature Generation using curves P-256 and P-384 with SHA-1, SHA2-224, SHA2-512, SHA3	Digital Signature Generation
ECDSA Signature Verification using curves P-256 and P-384 with SHA2-224, SHA2-512, SHA3	Digital Signature Verification
SHA2-224	Message Digest
SHA2-512	Message Digest
SM3	Message Digest
MD5	Message Digest
HMAC-SHA2-224	Message Authentication
HMAC-SHA2-512	Message Authentication
AES-CMAC	Message Authentication
Triple-DES	Message Authentication
Diffie-Hellman using all Diffie-Hellman Groups	Key Agreement
EC Diffie-Hellman using curves other than P-256 and P-384	Key Agreement
EC Diffie-Hellman using curves P-256 and P-384 Static Unified and OnePassDh schemes	Key Agreement
TLS KDF using MD5, SHA-1, SHA2-224, SHA2-512, SHA3	Key Derivation
SNMP KDF	Key Derivation
IKEv1 KDF	Key Derivation
IKEv2 KDF	Key Derivation

Table 6: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
AES-CBC	BC-UnAuth	Encryption/Decryption	Keys:128, 192, 256-bit keys with 128, 192, 256 bits of key strength	AES-CBC: (A4985, A5261)
AES-CTR	BC-UnAuth	Encryption/Decryption	Keys:128, 192, 256-bit keys with 128, 192, 256 bits of key strength	AES-CTR: (A4985, A5261)
AES-ECB	BC-UnAuth	Encryption/Decryption	Keys:128, 192, 256-bit keys with 128, 192, 256 bits of key strength	AES-ECB: (A4985, A5261)
AES-GCM (Authenticated Encryption/Decryption)	BC-Auth	Authenticated Encryption/Authenticated Decryption	Keys:128, 192, 256-bit keys with 128, 192, 256 bits of key strength	AES-GCM: (A4985, A5261)
AES-GMAC	MAC	Message authentication code (MAC)	Keys:128, 192, 256-bit keys with 128, 192, 256 bits of key strength	AES-GMAC: (A4985, A5261)
AES-GCM (Key Wrapping/Unwrapping)	KTS-Wrap	Key Wrapping, Key Unwrapping	Keys:128, 256-bit keys with 128 and 256 bits of key strength Compliance:Compliant with IG D.G	AES-GCM: (A4985, A5261)
AES-CBC with HMAC	KTS-Wrap	Key Wrapping, Key Unwrapping	Keys:128, 256-bit keys with 128 and 256 bits of key strength	AES-CBC: (A4985, A5261) HMAC-SHA2-256: (A4985, A5261) HMAC-SHA2-384: (A4985, A5261)
Counter DRBG	DRBG	Random Number Generation	Compliance:Compliant with SP800-90ARev1 Properties:with / without derivation function, prediction resistance disabled / enabled	Counter DRBG: (A4985, A5261)
RSA Signature Generation	DigSig-SigGen	Signature Generation	Keys:2048, 3072 and 4096-bit keys with 112 to 150 bits of key strength Hashes:SHA2-256, SHA2-384 Schemes:PKCS#1v1.5	RSA SigGen (FIPS186-5): (A4985, A5261)
RSA Signature Verification	DigSig-SigVer	Signature Verification	Keys: 2048, 3072 and 4096-bit keys with 112 to 150 bits of key strength Hashes:SHA2-256,	RSA SigVer (FIPS186-5): (A4985, A5261)

Name	Type	Description	Properties	Algorithms
			SHA2-384 Schemes:PKCS#1v1.5	
RSA Key Generation	AsymKeyPair-KeyGen CKG	Key Generation	Keys:2048, 3072 and 4096-bit keys with 112 to 150 bits of key strength Schemes:A.1.3 Random Probable Primes Compliance:IG D.H; SP800-133Rev2 section 5.1	RSA KeyGen (FIPS186-5): (A4985, A5261)
ECDSA Signature Generation	DigSig-SigGen	Signature Generation	Curves:P-256, P-384 with 128 and 192 bits of strength Hashes:SHA2-256, SHA2-384	ECDSA SigGen (FIPS186-5): (A4985, A5261)
ECDSA Signature Verification	DigSig-SigVer	Signature Verification	Curves:P-256, P-384 with 128 and 192 bits of strength Hashes:SHA2-256, SHA2-384	ECDSA SigVer (FIPS186-5): (A4985, A5261)
ECDSA Key Generation	AsymKeyPair-KeyGen CKG	Key Generation	Curves: P-256, P-384 with 128 and 192 bits of strength Schemes:FIPS 186-5, A.2.2 Rejection Sampling Compliance:IG D.H; SP800-133Rev2 sections 5.1 and 5.2	ECDSA KeyGen (FIPS186-5): (A4985, A5261)
ECDSA Key Verification	AsymKeyPair-KeyVer	Key Verification	Curves:P-256, P-384 with 128 and 192 bits of strength	ECDSA KeyVer (FIPS186-5): (A4985, A5261)
SHA	SHA	Message Digest		SHA-1: (A4985, A5261) SHA2-256: (A4985, A5261) SHA2-384: (A4985, A5261)
HMAC	MAC	Message authentication code (MAC)	Hashes:SHA-1, SHA2-256, SHA2-384 Keys:112 bits to 1024-bit keys with 112 to 256 bits of key strengths	HMAC-SHA-1: (A4985, A5261) HMAC-SHA2-256: (A4985, A5261) HMAC-SHA2-384: (A4985, A5261)
TLS Handshake	KAS-Full	Key agreement	Curves: P-256, P-384 with 128 and 192 bits of key strength	KAS-ECC-SSC Sp800-56Ar3: (A4985,

Name	Type	Description	Properties	Algorithms
			Compliance:IG D.F Scenario 2 (path 2)	A5261) TLS v1.2 KDF RFC7627: (A4985, A5261)
SSH Handshake	KAS-Full	Key agreement	EC Curves:P-256, P-384 with 128 and 192 bits of key strength Compliance:IG D.F Scenario 2 (path 2)	KDF SSH: (A4985, A5261) KAS-ECC-SSC Sp800-56Ar3: (A4985, A5261)
AES-CTR with HMAC	KTS-Wrap	Key Wrapping, Key Unwrapping	Keys:128, 256-bit keys with 128 and 256 bits of key strength	AES-CTR: (A4985, A5261) HMAC-SHA-1: (A4985, A5261) HMAC-SHA2-256: (A4985, A5261)

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES GCM IV

The User shall consider the following requirements and restrictions when using the module. AES-GCM IV is constructed in accordance with SP800-38D 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 shall be re-distributed. The AES GCM IV generation follows [RFC 5288] and shall only be 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 [SP800-52Rev2] section 3.3.1.

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	F5OS-A 1.7.0 on rSeries r12900 with Intel(R) Xeon(R) Platinum 8531N Ice Lake-SP	256 bits	256 bits	SHA3-256 (CAVP cert. #A3769)

Table 9: Entropy Sources

The module employs a Deterministic Random Bit Generator (DRBG) based on [SP800-90ARev1] 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 SP800-90B compliant entropy source specified in the *Entropy Sources* table to seed the DRBG.

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.

2.9 Key Generation

The module implements the following key generation methods:

- RSA, ECDSA asymmetric key generation, for digital signature schemes, compliant with [FIPS186-5], using an [SP800-90ARev1] DRBG for the generation of random values. These key generation services are compliant with section 5.1 of [SP800-133Rev2], according to IG D.H.
- EC Diffie-Hellman key generation using an [SP800-90ARev1] DRBG for the generation of random values. This key generation scheme is compliant with section 5.2 of [SP800-133Rev2], according to IG D.H.

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 provides the following key establishment services:

- EC Diffie-Hellman key agreement scheme compliant with SP800-56ARev3 and FIPS 140-3 IG D.F scenario 2 (path 2) is used as part of the TLS and SSH Protocols. The full EC Diffie-Hellman KAS implements a shared secret computation with the key derivation implemented by TLS v1.2 KDF RFC7627 (CVL) [SP 800-135] and KDF SSH (CVL) [SP 800-135]. EC Diffie-Hellman key agreement with P-256 and P-384 curves provides 128 and 192 bits of strength respectively.
- AES-GCM in the context of TLS protocol, with 128 and 256-bit keys, with 128 and 256 bits of key strength; compliant with IG D.G.
- AES-CBC or AES-CTR with HMAC in the context of SSH protocol, with 128 and 256-bit keys, with 128 and 256 bits of key strength; compliant with IG D.G.
- AES-CBC with HMAC in the context of TLS protocol, with 128 and 256-bit keys, with 128 and 256 bits of key strength; compliant with IG D.G.

2.11 Industry Protocols

The module implements the SSH key derivation function for use in the SSH protocol (RFC 4253 and 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.

For the purpose of the FIPS 140-3 validation, 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 switch to password authentication. Then the attacker probability of succeeding 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:

- 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 Command Line Interface (CLI) or Web Interface. However, 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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
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
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	RSA Signature Generation ECDSA 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
Create TLS Key	Create key for the SSL Certificate key file	Service Indicator: Approved	Key identification information	Confirmation of key creation	Counter DRBG RSA Key Generation ECDSA Key Generation	Administrator - TLS RSA public key: G - TLS RSA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						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: None - TLS RSA private key: None - TLS ECDSA public key: None - TLS ECDSA private key: None
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
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

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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- 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 - 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 ECDSA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						public key : Z - SSH ECDSA private key: Z - SSH EC Diffie-Hellman public key: Z - SSH EC Diffie-Hellman private key: Z - SSH shared secret: Z - SSH derived session key : Z - Password: Z - Intermediate key generation value: Z - Entropy input string : Z - DRBG seed : Z - DRBG internal state (V and key values): Z
Establish SSH session	SSH key exchange	SSH connection successful	Password; SSH ECDSA public key or SSH ECDSA private key	EC public key, Confirmation of SSH session establishment	ECDSA Signature Generation ECDSA Signature Verification ECDSA Key Generation ECDSA Key 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 - Intermediat

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						e key generation value: G,E,Z - SSH ECDSA public key : E - SSH ECDSA private key: E - Password: Resource Admin - 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 -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Intermediate key generation value: G,E,Z - SSH ECDSA public key : E - SSH ECDSA private key: E - Password: W,E
Maintain SSH Session (data encryption)	SSH data encryption	SSH connection successful	Plaintext	Ciphertext	AES-CBC AES-CTR AES-CBC with HMAC AES-CTR with HMAC	Administrator - SSH derived session key : E Resource Admin - SSH derived session key : E Operator - SSH derived session key : E
Maintain SSH Session (data decryption)	SSH data decryption	SSH connection successful	Ciphertext	Plaintext	AES-CBC AES-CTR AES-CBC with HMAC AES-CTR with HMAC	Administrator - SSH derived session key : E Resource Admin - SSH derived session key : E Operator - SSH derived session key : E
Maintain SSH Session (data integrity)	SSH data integrity	SSH connection successful	Message	MAC tag	HMAC	Administrator - SSH derived session key : E Resource Admin

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- 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 Operator - SSH EC Diffie-Hellman public key: Z - SSH EC Diffie-Hellman private key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Z - SSH shared secret: Z - SSH derived session key : Z
Establish TLS Session (key exchange)	Key exchange in TLS	Service Indicator: Approved	Ciphersuites	TLS EC Diffie-Hellman public key, Confirmation of establishment of TLS session	RSA Signature Generation RSA Signature Verification ECDSA Signature Generation ECDSA Signature Verification 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 RSA public key: E - TLS RSA private key: E - TLS ECDSA public key: E - TLS ECDSA private key: E Resource Admin - TLS EC Diffie-Hellman public key: E - TLS EC Diffie-Hellman private key: E - TLS pre-

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						primary secret : G,E - TLS primary secret: G,E - TLS derived session key : G - TLS RSA public key: E - TLS RSA private key: E Operator - TLS pre-primary secret : G,E - TLS primary secret: G,E - 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
Maintain TLS Session (authenticated data encryption)	TLS data encryption	Service Indicator: Approved	Ciphersuites, plaintext	Ciphertext	AES-GCM (Authenticated Encryption/Decryption) AES-CBC with HMAC HMAC	Administrator - TLS derived session key : E Resource Admin - TLS derived session key : E Operator - TLS derived session key : E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Maintain TLS Session (authenticated data decryption)	TLS data decryption	Service Indicator: Approved	Ciphersuites, ciphertext	Plaintext	AES-GCM (Authenticated Encryption/Decryption) AES-CBC with HMAC	Administrator - TLS derived session key : E Resource Admin - TLS derived session key : E Operator - TLS derived session key : E
Maintain TLS Session (data authentication)	TLS data authentication	Service Indicator: Approved	Ciphersuites, message	MAC tag	HMAC	Administrator - TLS derived session key : E Resource Admin - TLS derived session key : E Operator - 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 derived session key : Z Resource

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						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 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 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	None	Administrator Resource

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
				the show status command		Admin Operator
Self-test	Execute integrity test; Execute the CASTs	None	None	Pass/ fail results of self-tests	AES-CBC AES-CTR AES-ECB AES-GCM (Authenticated Encryption/Decryption) AES-GMAC AES-GCM (Key Wrapping/Unwrapping) AES-CBC with HMAC Counter DRBG RSA Signature Generation RSA Signature Verification RSA Key Generation ECDSA Signature Generation ECDSA Signature Verification ECDSA Key Generation ECDSA Key Verification SHA HMAC TLS Handshake SSH Handshake AES-CTR with HMAC	Administrator Resource Admin Operator
Show tenant	Lists tenant information	None	None	Lists tenant information	None	Administrator Resource Admin Operator

Table 13: Approved Services

The environment variable SECURITY_FIPS140_CIPHER_STRICT is exported with the cipher restriction status. If the cipher_restricted status is enabled, the status output from the service indicator is returned in the /var/log/audit.log file. Using an approved service will provide an indicator which shows which approved algorithms were used. If the cipher_restricted status is disabled, there is no service indicator output.

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

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

Name	Description	Algorithms	Role
Establish TLS session (signature generation and verification)	Signature generation and verification	DSA EdDSA Signature Generation and Verification using Ed25519 RSA PKCS#1 v1.5 Signature Generation and Verification using 2048, 3072 or 4096-bits modulus with SHA-1, SHA2-224, SHA2-512 RSA PKCS #1 v1.5 Signature Generation and Verification with modulus other than 2048, 3072 or 4096 bits, for all SHA sizes RSA PSS Signature Generation and Verification using 2048, 3072 or 4096-bits modulus ECDSA Signature Generation and Verification using curves other than P-256 and P-384, all SHA sizes ECDSA Signature Generation using curves P-256 and P-384 with SHA-1, SHA2-224, SHA2-512, SHA3 ECDSA Signature Verification using curves P-256 and P-384 with SHA2-224, SHA2-512, SHA3	User/CO
Establish TLS session (key exchange)	Key exchange	RSA Asymmetric Encryption and Decryption Diffie-Hellman using all Diffie-Hellman Groups EC Diffie-Hellman using curves other than P-256 and P-384 EC Diffie-Hellman using curves P-256 and P-384 Static Unified and OnePassDh schemes TLS KDF using MD5, SHA-1, SHA2-224, SHA2-512, SHA3	User/CO
Maintain TLS session (data encryption)	Data encryption	AES-CCM AES-CFB AES-OFB AES-KW DES RC4 Triple-DES SM2 SM4	User/CO
Maintain TLS session (data authentication)	Data authentication	HMAC-SHA2-224 HMAC-SHA2-512 AES-CMAC Triple-DES	User/CO
Create TLS key	Key generation	RSA Key Generation with modulus sizes other than 2048, and 4096-bits ECDSA Key Generation and Verification with curves other than P-256 and P-384 Safe Primes Key Generation and Verification for Diffie-Hellman	User/CO

Name	Description	Algorithms	Role
Message digest	Message digest	SHA2-224 SHA2-512 SM3 MD5	User/CO
Key derivation	Key derivation	SNMP KDF IKEv1 KDF IKEv2 KDF	User/CO

Table 14: Non-Approved Services

4.5 External Software/Firmware Loaded

Not applicable.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified using the approved integrity technique HMAC-SHA-384, as listed in the section 10.1, by comparing the HMAC-SHA-384 checksum values of the installed binaries calculated at run time with the stored values computed at build time. If the values do not match, the module enters the Error state.

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

6.1 Operational Environment Type and Requirements

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). The Crypto Officer shall be responsible for the storage of the label kits.

Table 15: Mechanisms and Actions Required

The module tested on the platform listed in the *Tested Operational Environments - Software, Firmware, Hybrid* table is enclosed in a hard-metallic production grade enclosure that provides opacity and prevents visual inspection of internal components. Each test platform is fitted with tamper evident labels to provide physical evidence of attempts to gain access inside the enclosure. The tamper evident labels shall be installed for the module to operate in approved mode of operation.

7.2 User Placed Tamper Seals

Number: 5

Placement: Shown in Figure 3

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 48 hours.

Operator Responsible for Securing Unused Seals: Crypto Officer

Part Numbers: F5-ADD-BIG-FIPS140

The pictures below show the location of all tamper-evident labels for the platform.

Figure 3 - Tamper labels on r12900

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 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 memory occupied by SSPs is allocated by regular memory allocation operating system calls.	Dynamic

Table 16: Storage Areas

SSPs are only accessible to the authenticated operator, to which the SSPs are associated.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
TLS/SSH Protocol Output	CM Software	GPC EXT using a network port	Plaintext	Automated	Electronic	TLS Handshake
Service Input	CM Software via EXT Path	CM Software	Plaintext	Automated	Electronic	

Table 17: SSP Input-Output Methods

During the TLS handshake, the keys that are entered or output to the module over the network includes RSA/ECDSA public keys. For TLS with EC Diffie-Hellman key exchange, the TLS pre-primary secret is established during key agreement and is not output from the module. Once the TLS session is established, any key or data transfer performed thereafter is protected by authenticated encryption mode using AES-GCM or by AES encryption and HMAC authentication through a mutually agreed AES and HMAC session keys derived by applying TLS v1.2 KDF RFC7627 (CVL) [SP800-135].

For SSH with EC Diffie-Hellman key exchange, the SSH shared secret is established during key agreement and is not output from the module. SSH ECDSA public keys can be imported into the module by the CO using the "Configure SSH user configuration" service. Once the SSH session is established, any key or data transfer performed thereafter is protected by AES encryption and HMAC authentication through a mutually agreed AES and HMAC session keys derived by applying KDF SSH (CVL) [SP800-135].

There are no encrypted SSPs that are directly entered.

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 Crypto Officer 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 Crypto Officer calling Reboot System service

Zeroization Method	Description	Rationale	Operator Initiation
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 18: SSP Zeroization Methods

The zeroization methods listed in the *SSP Zeroization Methods* table, overwrite the memory occupied by keys with "zeros" or pre-defined values.

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	2048, 3072, 4096 bits - 112-150 bits	Asymmetric - PSP	RSA Key Generation		RSA Signature Verification
TLS RSA private key	RSA private key pair used for digital signature generation in TLS protocol	2048, 3072, 4096 bits - 112-150 bits	Asymmetric - CSP	RSA Key Generation		RSA Signature Generation
TLS ECDSA public key	ECDSA public key used for digital signature verification in TLS protocol	P-256, P-384 - 128 and 192 bits	Asymmetric - PSP	ECDSA Key Generation		ECDSA Signature Verification
TLS ECDSA private key	ECDSA private key used for digital signature generation in TLS protocol	P-256, P-384 - 128 and 192 bits	Asymmetric - CSP	ECDSA Key Generation		ECDSA Signature Generation
TLS EC Diffie-Hellman public key	ECDH public key used in TLS protocol key exchange	P-256, P-384 - 128 and 192 bits	Asymmetric - PSP	ECDSA Key Generation		TLS Handshake
TLS EC Diffie-Hellman private key	ECDH private key used in TLS protocol key exchange	P-256, P-384 - 128 and 192 bits	Asymmetric - CSP	ECDSA Key Generation		TLS Handshake
TLS pre-primary secret	TLS pre-primary secret used for deriving the TLS primary secret	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	256 bits - 256 bits	Pre-primary secret - CSP		TLS Handshake	TLS Handshake
TLS derived session key	TLS derived session key derived from TLS primary secret	AES: 128 and 256 bits; HMAC: 112-256 bits - 112-256 its	Symmetric - CSP	TLS Handshake		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SSH ECDSA public key	ECDSA public key used for SSH key-based authentication	P-256, P-384 - 128 and 192 bits	Asymmetric - PSP			ECDSA Signature Verification
SSH ECDSA private key	ECDSA private key used for SSH key-based authentication	P-256, P-384 - 128 and 192 bits	Asymmetric - CSP			ECDSA Signature Generation
SSH EC Diffie-Hellman public key	EC Diffie-Hellman public key used for SSH handshake	P-256, P-384 - 128 and 192 bits	Asymmetric - PSP	ECDSA Key Generation		SSH Handshake
SSH EC Diffie-Hellman private key	EC Diffie-Hellman private key used for SSH handshake	P-256, P-384 - 128 and 192 bits	Asymmetric - CSP	ECDSA Key Generation		SSH Handshake
SSH shared secret	SSH shared secret established during SSH shared secret computation	P-256, P-384 - 128 and 192 bits	Shared secret - CSP		SSH Handshake	SSH Handshake
SSH derived session key	SSH derived session key derived from SSH shared secret	AES: 128 and 256 bits; HMAC: 112-256 bits - 112-256 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	256 bits - 256 bits	Entropy - CSP			Counter DRBG
DRBG seed	DRBG seed derived from the entropy input string	256 bits - 256 bits	Seed - CSP	Counter DRBG		Counter DRBG
DRBG internal state (V and key values)	DRBG internal state derived from the DRBG seed	256 bits - 256 bits	Internal state - CSP	Counter DRBG		Counter DRBG
Intermediate key generation value	Temporary value generated during key pair generation services	256-4096 bits - 112-256 bits	Intermediate value - CSP	RSA Key Generation ECDSA Key Generation		RSA Key Generation ECDSA Key Generation TLS Handshake SSH Handshake

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS RSA public key		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		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
TLS EC Diffie-Hellman public key	TLS/SSH Protocol Output	RAM:Plaintext	From service invoked to service completed or module reboot	Secure Erase 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	Secure Erase 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	Secure Erase Reboot System Closing TLS/SSH Connection	TLS primary secret:Derives
TLS primary secret		RAM:Plaintext	From service invoked to service completed or module reboot	Secure Erase Reboot System Closing TLS/SSH Connection	TLS pre-primary secret :Derived From TLS derived session key :Derives
TLS derived session key		RAM:Plaintext	From service invoked to service completed or module reboot	Secure Erase Reboot System Closing TLS/SSH Connection	TLS primary secret:Derived From
SSH ECDSA public key	Service Input	SSD :Plaintext	From service invoked to service completed or module reboot	Secure Erase	SSH ECDSA private key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
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	TLS/SSH Protocol Output	RAM:Plaintext	From service invoked to service completed or module reboot	Secure Erase 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	Secure Erase 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	Secure Erase Reboot System Closing TLS/SSH Connection	SSH EC Diffie-Hellman public key:Established By SSH EC Diffie-Hellman private key:Established By SSH derived session key :Derives
SSH derived session key		RAM:Plaintext	From service invoked to service completed or module reboot	Secure Erase Reboot System Closing TLS/SSH Connection	SSH shared secret:Derived From
Password	Service Input	SSD :Plaintext	N/A	Secure Erase	
Entropy input string		RAM:Plaintext	Storage duration during the usage of the CSP	Secure Erase Reboot System	DRBG seed :Derives
DRBG seed		RAM:Plaintext	Storage duration during the usage of the CSP	Secure Erase 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	Secure Erase Reboot System	DRBG seed :Derived From
Intermediate key generation value		RAM:Plaintext	From service invoked to service completed or module reboot	Secure Erase 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

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					SSH EC Diffie-Hellman private key:Derives TLS EC Diffie-Hellman public key:Derives TLS EC Diffie-Hellman private key:Derives

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, 2030.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-384 (A4985)	SHA2-384	Message authentication	SW/FW Integrity	Module is operational and services are available for use	Integrity test for F5OS-A version 1.7.0
HMAC-SHA2-384 (A5261)	SHA2-384	Message authentication	SW/FW Integrity	Module is operational and services are available for use	Integrity test for F5OS-A version 1.7.0

Table 21: Pre-Operational Self-Tests

The pre-operational self-test is performed automatically when the module is powered on. At initialization, the module performs pre-operational self-test (integrity test) and the CASTs. Services are not available, and the data output is inhibited during the pre-operational self-test. Upon successful completion of the pre-operational self-tests and CASTs, the module becomes operational and cryptographic services are available for use. If the module fails any of the self-tests, the module returns an error code, and transitions to the Error state.

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.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A4985)	256-bit key	Encrypt KAT	CAST	Module is operational and services are available for use	Encryption	Module initialization
AES-GCM (A5261)	256-bit key	Encrypt KAT	CAST	Module is operational and services are available for use	Encryption	Module initialization
AES-GCM (A4985)	256-bit key	Decrypt KAT	CAST	Module is operational and services are available for use	Decryption	Module initialization
AES-GCM (A5261)	256-bit key	Decrypt KAT	CAST	Module is operational and services are available for use	Decryption	Module initialization
AES-ECB (A4985)	128 bit-key	Encrypt KAT	CAST	Module is operational and services are available for use	Encryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A5261)	128 bit-key	Encrypt KAT	CAST	Module is operational and services are available for use	Encryption	Module initialization
AES-ECB (A4985)	128 bit-key	Decrypt KAT	CAST	Module is operational and services are available for use	Decryption	Module initialization
AES-ECB (A5261)	128 bit-key	Decrypt KAT	CAST	Module is operational and services are available for use	Decryption	Module initialization
RSA SigGen (FIPS186-5) (A4985)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	Sign KAT	CAST	Module is operational and services are available for use	Signature generation	Module initialization
RSA SigGen (FIPS186-5) (A5261)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	Sign KAT	CAST	Module is operational and services are available for use	Signature generation	Module initialization
RSA SigVer (FIPS186-5) (A4985)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	Verify KAT	CAST	Module is operational and services are available for use	Signature verification	Module initialization
RSA SigVer (FIPS186-5) (A5261)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	Verify KAT	CAST	Module is operational and services are available for use	Signature verification	Module initialization
ECDSA SigGen (FIPS186-5) (A4985)	P-256 and SHA2-256	Sign KAT	CAST	Module is operational and services are available for use	Signature generation	Module initialization
ECDSA SigGen (FIPS186-5) (A5261)	P-256 and SHA2-256	Sign KAT	CAST	Module is operational and services are available for use	Signature generation	Module initialization
ECDSA SigVer (FIPS186-5) (A4985)	P-256 and SHA2-256	Verify KAT	CAST	Module is operational and services are available for use	Signature verification	Module initialization
ECDSA SigVer (FIPS186-5) (A5261)	P-256 and SHA2-256	Verify KAT	CAST	Module is operational and services are available for use	Signature verification	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KAS-ECC-SSC Sp800-56Ar3 (A4985)	P-256	Shared secret computation	CAST	Module is operational and services are available for use	Shared secret computation	Module initialization
KAS-ECC-SSC Sp800-56Ar3 (A5261)	P-256	Shared secret computation	CAST	Module is operational and services are available for use	Shared secret computation	Module initialization
HMAC-SHA-1 (A4985)	SHA-1	HMAC KAT	CAST	Module is operational and services are available for use	Message authentication	Module initialization
HMAC-SHA-1 (A5261)	SHA-1	HMAC KAT	CAST	Module is operational and services are available for use	Message authentication	Module initialization
HMAC-SHA2-256 (A4985)	SHA2-256	HMAC KAT	CAST	Module is operational and services are available for use	Message authentication	Module initialization
HMAC-SHA2-256 (A5261)	SHA2-256	HMAC KAT	CAST	Module is operational and services are available for use	Message authentication	Module initialization
HMAC-SHA2-384 (A4985)	SHA2-384	HMAC KAT	CAST	Module is operational and services are available for use	Message authentication	Module initialization
HMAC-SHA2-384 (A5261)	SHA2-384	HMAC KAT	CAST	Module is operational and services are available for use	Message authentication	Module initialization
KDF SSH (A4985)	SHA2-256	SSH KDF KAT	CAST	Module is operational and services are available for use	Key derivation	Module initialization
KDF SSH (A5261)	SHA2-256	SSH KDF KAT	CAST	Module is operational and services are available for use	Key derivation	Module initialization
KDF TLS (A4985)	SHA2-256	TLS 1.2 KDF KAT	CAST	Module is operational and services are available for use	Key derivation	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
TLS v1.2 KDF RFC7627 (A5261)	SHA2-256	TLS 1.2 KDF KAT	CAST	Module is operational and services are available for use	Key derivation	Module initialization
ECDSA KeyGen (FIPS186-5) (A4985)	SHA2-256 and respective curve	ECDSA PCT	PCT	Successful key pair generation	Signature Generation & Signature Verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A5261)	SHA2-256 and respective curve	ECDSA PCT	PCT	Successful key pair generation	Signature Generation & Signature Verification	Key pair generation
RSA KeyGen (FIPS186-5) (A4985)	SHA2-256 and respective keys	RSA PCT	PCT	Successful key pair generation	Signature Generation & Signature Verification	Key pair generation
RSA KeyGen (FIPS186-5) (A5261)	SHA2-256 and respective keys	RSA PCT	PCT	Successful key pair generation	Signature Generation & Signature Verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A4985)	SHA2-256 and respective curve	ECDH PCT	PCT	Successful key pair generation	Signature generation & signature verification PCT that covers key pair generation for EC Diffie-Hellman	Key pair generation
ECDSA KeyGen (FIPS186-5) (A5261)	SHA2-256 and respective curve	ECDH PCT	PCT	Successful key pair generation	Signature generation & signature verification PCT that covers key pair generation for EC Diffie-Hellman	Key pair generation
Counter DRBG (A4985)	AES-256 with/without derivation function; Health test per section 11.3 of SP 800-90ARev1	CTR_DRBG KAT	CAST	Module is operational and services are available for use	KAT CTR_DRBG with AES with 256-bit key with and without prediction resistnace	Module initialization
Counter DRBG (A5261)	AES-256 with/without derivation function; Health test per section 11.3 of SP 800-90ARev1	CTR_DRBG KAT	CAST	Module is operational and services are available for use	KAT CTR_DRBG with AES with 256-bit key with and without prediction resistnace	Module initialization
Entropy Source	Health Test	RCT	CAST	Module is operational and services are available for use	Health test at start-up: performed on 1,024 consecutive samples	Module initialization
Entropy Source	Health Test	RCT	CAST	Module is operational	Health test during runtime: performed	Continuously

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				and services are available for use	on 1,024 consecutive samples	
Entropy Source	Health Test	APT	CAST	Module is operational and services are available for use	Health test during start-up: performed on 512 consecutive samples	Module initialization
Entropy Source	Health Test	APT	CAST	Module is operational and services are available for use	Health test during runtime: performed on 512 consecutive samples	Continuously

Table 22: Conditional Self-Tests

During the conditional self-tests, services are not available 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 (A4985)	Message authentication	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-384 (A5261)	Message authentication	SW/FW Integrity	On Demand	Manually

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (A4985)	Encrypt KAT	CAST	On Demand	Manually
AES-GCM (A5261)	Encrypt KAT	CAST	On Demand	Manually
AES-GCM (A4985)	Decrypt KAT	CAST	On Demand	Manually
AES-GCM (A5261)	Decrypt KAT	CAST	On Demand	Manually
AES-ECB (A4985)	Encrypt KAT	CAST	On Demand	Manually
AES-ECB (A5261)	Encrypt KAT	CAST	On Demand	Manually
AES-ECB (A4985)	Decrypt KAT	CAST	On Demand	Manually
AES-ECB (A5261)	Decrypt KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A4985)	Sign KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5261)	Sign KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A4985)	Verify KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5261)	Verify KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A4985)	Sign KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigGen (FIPS186-5) (A5261)	Sign KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A4985)	Verify KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5261)	Verify KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4985)	Shared secret computation	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A5261)	Shared secret computation	CAST	On Demand	Manually
HMAC-SHA-1 (A4985)	HMAC KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A5261)	HMAC KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4985)	HMAC KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A5261)	HMAC KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A4985)	HMAC KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A5261)	HMAC KAT	CAST	On Demand	Manually
KDF SSH (A4985)	SSH KDF KAT	CAST	On Demand	Manually
KDF SSH (A5261)	SSH KDF KAT	CAST	On Demand	Manually
KDF TLS (A4985)	TLS 1.2 KDF KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A5261)	TLS 1.2 KDF KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A4985)	ECDSA PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5261)	ECDSA PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A4985)	RSA PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A5261)	RSA PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A4985)	ECDH PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5261)	ECDH PCT	PCT	On Demand	Manually
Counter DRBG (A4985)	CTR_DRBG KAT	CAST	On Demand	Manually
Counter DRBG (A5261)	CTR_DRBG KAT	CAST	On Demand	Manually
Entropy Source	RCT	CAST	On Demand	Manually
Entropy Source	RCT	CAST	Continuously	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Entropy Source	APT	CAST	On Demand	Manually
Entropy Source	APT	CAST	Continuously	Manually

Table 24: 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 CAST Failure of any of the PCTs Failure of the APT, RCT at runtime Failure of the APT, RCT at restart (power on)	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 25: Error States

All data output and cryptographic operations are inhibited when the module is in the Error State.

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 1.7.0 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 and non-Approved modes of operation are 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

The approved and non-approved security functions available to users are listed in section 2, the physical ports, and logical interfaces available to users are specified in section 3.1. The Approved and non-Approved modes of operation are specified in section 2.4. The algorithm-specific information is listed in section 2.7.

12 Mitigation of Other Attacks

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

