F5, Inc.

F5OS-A Cryptographic Module

Module Version: 1.5.1

FIPS Security Level 2

FIPS 140-3 Non-Proprietary Security Policy

Document Version 1.2

Last update: November 2024

Prepared by:

atsec information security corporation

4516 Seton Center Parkway, Suite 250

Austin, TX 78759

www.atsec.com

Table of Contents

Copyrights and Trademarks

F5®, BIG-IP® are registered trademarks of F5, Inc.

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

1 General

This document is the non-proprietary FIPS 140-3 Security Policy for the F5OS-A Cryptographic Module with firmware version 1.5.1. 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.

This document provides all tables and diagrams (when applicable) required by NIST SP 800-140B.

ISO/IEC 24759 Section 6. [Number Below]	FIPS 140-3 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

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: Multi Chip Standalone

2.2 Operating Environments

Operating system	Hardware Platform	Processors	PAA/ Acceleration
F5OS-A 1.5.1	r4800	Intel® Atom® P5342 Snow Ridge NS	with and without PAA
F5OS-A 1.5.1	r5900	Intel® Xeon® Silver 4314 Ice Lake-SP	with and without PAA
F5OS-A 1.5.1	r5920-DF	Intel® Xeon® Silver 4314 Ice Lake-SP	with and without PAA
F5OS-A 1.5.1	r10900	Intel® Xeon® Gold 6312U Ice Lake-SP	with and without PAA
F5OS-A 1.5.1	r10920-DF	Intel® Xeon® Gold 6312U Ice Lake-SP	with and without PAA

Table 2 - Tested Operational Environments

2.3 Modes of Operation

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 pre-operational self-tests and CASTs succeed. The module automatically switches between the approved and non-approved modes depending on the services requested by the operator. The status indicator of the mode of operation is equivalent to the indicator of the service that was requested. SSPs used or stored in the Approved mode are not used in the non-Approved mode, and vice versa.

In the Approved Mode, the cryptographic module provides the following cryptographic algorithms whose CAVP certificates are in Table 3 below. Not all the ACVP tested capabilities are used by the module in approved mode of operation.

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s)/ Key Strength(s)	Use / Function
A3896, A5260	AES [FIPS 197, SP800-38A, SP800-38C, SP800 38D]	ECB, CBC, GCM, CTR	128 / 192 / 256-bit keys with key strengths from 128 to 256 bits	Encryption and Decryption
A3896, A5260	KTS (AES) [FIPS 197, SP800-38D, SP800- 38F]	GCM	128 / 256-bit AES keys with key strengths 128 or 256 bits	Key Wrapping / Unwrapping
A3896, A5260		AES-CBC key and HMAC-SHA-1, HMAC-SHA2-256, or HMAC-SHA2-384	128 / 256-bit AES and HMAC keys with key strengths 128 or 256 bits	
A3896, A5260		AES-CBC/ AES-CTR keys and HMAC-SHA-1, HMAC-SHA2-256	128 / 256-bit AES and HMAC keys with key strengths from 128 or 256 bits	
A3896, A5260	AES [FIPS 197, SP800-38B, SP800 38D]	GMAC	128 / 192 / 256-bit AES keys with key strengths from 128 and 256 bits	MAC Generation and Verification
A3896, A5260	CTR_DRBG [SP800-90Ar1]	AES 256 in CTR mode, with / without derivation function, prediction resistance disabled / enabled	Entropy input string (256-bits), V (128-bits) and key (256-bits) values	Random Number Generation
A3896, A5260	RSA [FIPS 186-4]	B.3.3 Random Probable Primes	2048, 3072 and 4096-bit keys with key strengths 112 to 150-bits	RSA key generation
		PKCS#1v1.5: SHA2-256, SHA2-384	2048, 3072 and 4096-bit keys with key strengths 112 to 150-bits	RSA signature generation
		PKCS#1v1.5: SHA2-256, SHA2-384	2048, 3072 and 4096-bit keys with key strengths 112 to 150-bits	RSA signature verification
A3896, A5260	ECDSA [FIPS 186-4]	B.4.2 Testing Candidates	P-256 and P-384 with key strengths 128 and 192-bits	ECDSA key pair generation / verification
		SHA2-256, SHA2-384	P-256 and P-384 with key strengths 128 and 192-bits	ECDSA signature generation and verification
A3896, A5260	SHS [FIPS180-4]	SHA-1 SHA2-256 SHA2-384	N/A	Message digest
A3896, A5260	HMAC [FIPS 198-1]	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384	112 bits to 1024-bits with key strengths 112 to 256-bits	Message authentication

CAVP Cert	Algorithm and Standard	Mode / Method	Description / Key Size(s)/ Key Strength(s)	Use / Function
A3896, A5260	KAS-ECC-SSC [SP800-56Ar3]	Ephemeral Unified: KAS Role: initiator, responder	P-256, P-384 with key strengths 128 and 192-bits	Shared Secret Computation used in Key Agreement Scheme (KAS) IG D.F scenario 2 (path 2)
A3896, A5260	SSH KDF[1] (CVL) [SP800-135]	AES-128, AES-256 with SHA2-256, SHA2-384	256-bit keys with 256-bits key strength	Key derivation
A3896, A5260	TLS KDF[1] (CVL) [SP800-135, RFC 7627]	TLS v1.2	256-bits	Key derivation
(vendor affirmed)	CKG [SP800-133r2] CTR_DRBG [SP800-90Ar1] KAS-ECC-SSC [SP800-56Ar3] RSA, ECDSA [FIPS 186-4]	DRBG produces the random numbers for key generation of RSA, ECDSA and EC Diffie-Hellman	RSA Sizes: 2048, 3072 and 4096-bits key with 112 and 150-bits key strength ECDSA and EC Diffie-Hellman: P-256 and P-384 with 128 and 192-bits key strength	Key generation

Table 3 - Approved Algorithms

The module does not implement any non-approved algorithms allowed in the approved mode of operation with or without security claimed.

The following table lists the non-approved algorithms not allowed in approved mode along with their usage.

Algorithm/ Functions	Use/Function
AES modes: CCM, CFB, OFB, XTS and KW modes; DES; RC4; Triple-DES; SM2, SM4	Symmetric Encryption and Decryption
RSA	Asymmetric Encryption and Decryption
RSA Key generation	with modulus size other than 2048, 3072 and 4096-bits;
DSA	domain parameter generation, domain parameter verification, Key pair generation
DSA digital signature	Signature generation and verification using any key size

---

[1] No parts of the TLS / SSH protocols except the KDF has been reviewed or tested by the CAVP and CMVP

Algorithm/ Functions	Use/Function
EdDSA digital signature	Signature generation and verification using Ed25519
ECDSA Key generation/ verification	With curves other than P-256 and P-384
Safe Primes Key generation/verificatio n	Key generation for Diffie-Hellman using any safe prime groups
RSA digital signature	- Signature Generation: PKCS#1 v1.5 using 2048, 3072 or 4096-bits modulus with SHA-1, SHA2-224, SHA2-512 - Signature Verification PKCS#1 v1.5 using 2048, 3072 or 4096-bits modulus with SHA-1, SHA2-224, SHA2-512 - Signature Generation and Verification using PKCS #1 v1.5 scheme with modulus other than 2048, 3072 or 4096 bits, for all SHA sizes - Signature Generation PSS using 2048, 3072 or 4096-bits modulus - Signature Verification PSS using 2048, 3072 or 4096-bits modulus - Signature Generation and Verification using Probabilistic Signature Scheme (PSS) specified in ANSI X9.31 standard - modulus sizes other than 2048, 3072 and 4096-bits
ECDSA digital signature	- Digital Signature Generation and Verification using curves other than P-256 and P-384, all SHA sizes - Digital Signature Generation using curves P-256 and P-384 with SHA-1, SHA2-224, SHA2-512, SHA3 - Digital Signature Verification using curves P-256 and P-384 with SHA2-224, SHA2-512, SHA3
SHA2-224 SHA2-512 SM3 MD5	Message Digest
HMAC-SHA2-224 HMAC-SHA2-512 AES-CMAC Triple-DES	Message Authentication

Diffie-Hellman
EC Diffie-Hellman
	Key Agreement Scheme:
	    - 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
SNMP KDF,
IKEv1, IKEv2 KDF
	Key Derivation function in the context of:
	    - TLS using MD5 / SHA-1 / SHA2-224 / SHA2-512 / SHA3
	    - SNMP using any SHA variant
	    - IKE using any SHA variant

Table 4 - Non-Approved Algorithms Not Allowed in the Approved Mode of Operation

2.4 Hardware Platform Photographs

Figures below show the various platforms on which the module was tested.

Figure 1 - r4800 isometric view

Figure 2 - r5900 front view

Figure 3 - r5920-DF front view

Figure 4 – r10900, r10920-DF front view (same chassis for the test platforms)

2.5 Block Diagram and Cryptographic Boundary Descriptions

The module cryptographic boundary is defined by the red dotted line in Figure 5. The TOEPP is defined by the tested platforms listed in Table 2 and delineated by the black rectangle in Figure 5. Figure 5 also depicts the flow of status output (SO), control input (CI), data input (DI) and data output (DO) interfaces.
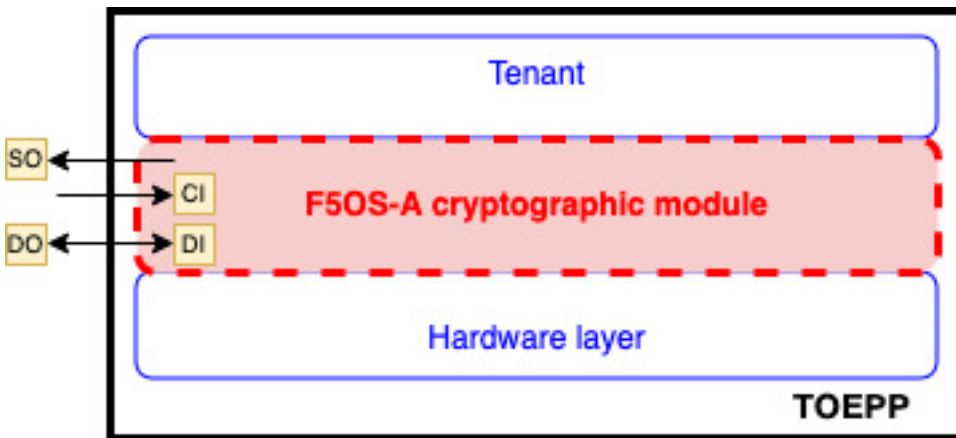
Figure 5 - Block Diagram

3 Cryptographic Module 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 can be used for data input, data output, status output or control input.

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 it runs.

Physical port	Logical Interface[2]	Data that passes over port/interface
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.
Power interface	Power Input	Power

Table 5 - Ports and Interfaces

---

[2] The module does not implement Control Output interface.

4 Roles, Services, and Authentication

4.1 Roles

The module supports one CO role and one User role. Maintenance role is not supported. The FIPS 140-3 roles are defined below and corresponding service with input and output are described in Table 6.

- Crypto Officer (CO) role: The Crypto Officer is represented by the administrator/root of the module. This entity performs module installation and initialization. This role has full access to the system and can create, delete, and manage other User roles on the system. At initialization, the CO is the only available role and only the CO can create the user roles.
- The FIPS140-3 User role is mapped to multiple module roles: Operator, Resource Admin and, Tenant-console.

The list of services available to the CO and user roles are defined in Table 8 and Table 9.

FIPS 140-3 Role	Module Role	Service	Input	Output
CO User	Administrator Resource Admin Operator	List users	None	List of user accounts
CO	Administrator	Create additional User	Username / password	Confirmation of account creation
CO	Administrator	Modify existing Users	Username / modification (new username, role, password expiry date/tally count)	Confirmation of account modification
CO	Administrator	Delete User	Username	Confirmation of deletion
CO	Administrator	Unlock User	Username	Confirmation of unlock
CO User	Administrator Resource Admin Operator	Update own password	Own password	Confirmation of update of password
CO	Administrator	Update others password	Username / password	Confirmation of update
CO	Administrator	Configure password policy	New password policy	Confirmation of configuration change
CO User	Administrator Resource Admin	Create TLS certificate	Certificate identification information	Confirmation of certificate creation
CO User	Administrator Resource Admin	Create TLS Key	Key identification information	Confirmation of key creation
CO User	Administrator Resource Admin	Delete TLS Certificate/Key	Key identification information	Confirmation of key deletion
CO User	Administrator Resource Admin	List Certificate	List of certificates to display	Certificate expiration information
CO User	Administrator Resource Admin	List private keys	List of private keys to display	List of private keys
CO User	Administrator Resource Admin	View System Audit Log	N/A	Display of system audit logs
CO User	Administrator Resource Admin	Configure SSH access options	SSH access / IP address list	Confirmation of configuration of SSH access options

FIPS 140-3 Role	Module Role	Service	Input	Output
CO	Administrator	Configure SSH user configuration	SSH ECDSA key pair (public)	Confirmation of configuration of SSH user configuration
CO User	Administrator Resource Admin	Create a tenant	password / tenant console role	Confirmation of the tenant-console role
User	Tenant-console	Connecting to tenant-console via SSH	F5 rSeries platform management address / tenant- console / password	Confirmation of Access to the tenant-console remotely over SSH
User	Tenant-console	Closing the tenant-console SSH session	N/A	Confirmation of tenant-console SSH session closure
CO	Administrator	Reboot System	N/A	Confirmation of system reboot
CO	Administrator	Secure Erase	Selection option	Confirmation of full system zeroization
CO User	Administrator Resource Admin Operator	SSH session service	User / address / password / algorithms / key sizes/ primary secret	Confirmation of SSH session establishment
CO User	Administrator Resource Admin Operator	Closing SSH Session	N/A	Confirmation of SSH session closure
CO User	Administrator Resource Admin Operator	TLS session service	Address / algorithms/ keys	Confirmation of establishment of TLS session
CO User	Administrator Resource Admin Operator	Closing TLS session	N/A	Confirmation of TLS session closure
CO User	Administrator Resource Admin Operator	Show version	None	Version information, and module name
CO User	Administrator Resource Admin Operator	Show license	None	FIPS license information
CO User	Administrator Resource Admin Operator	Show status	None	Status of the specific service passed in the show status command
CO User	Administrator Resource Admin Operator	Self- test	None	Pass/ fail results of self-tests
CO User	Administrator Resource Admin Operator	Show tenant	None	Lists tenant information

Table 6 - Roles, Service Commands, Input and Output

4.2 Authentication

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.

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).

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

Table 7 lists the required role-based authentication method for the Crypto Office role and the User role depending upon which interface is being used.

Role	Authentication Method	Authentication Strength
Crypto Officer User	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.
Crypto Officer User	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.

Table 7 - Roles and Authentication

4.3 Services

Table 8 lists the Approved services, the service name, description, the Approved security function being used by the service, the keys and SSPs accessed by the service, the roles used by the service, access rights to keys and SSPs and the FIPS 140-3 service indicator returned by the service.

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

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
User Management Services						
List users	Display list of all user accounts	N/A	N/A	CO User	N/A	None
Create additional User	Create additional user	N/A	password	CO	W	None
Modify existing Users	Modify existing users	N/A	N/A	CO	N/A	None
Delete User	Delete existing user	N/A	N/A	CO	N/A	None
Unlock User	Remove lock from user who has exceeded login attempts	N/A	N/A	CO	N/A	None
Update own password	Update own password	N/A	password	CO User	W	None
Update others password	Update others password	N/A	password	CO	W	None
Configure Password Policy	Set password policy features	N/A	N/A	CO	N/A	None
Certificate and Key Management Services						
Create TLS Certificate	Self-signed certificate creation	RSA / ECDSA SigGen	TLS RSA Public / Private keys RLS ECDSA Public / Private keys	CO User	E	Service Indicator: Approved
Create TLS Key	Used for the SSL Certificate key file	RSA / ECDSA KeyGen CTR_DRBG CKG	TLS RSA Public / Private keys TLS ECDSA Public / Private keys	CO User	G	Service Indicator: Approved
			DRBG seed		E	
			DRBG internal state (V and key values)		W, E	

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Delete TLS Certificate /Key	Self-signed certificate / key deletion	N/A	TLS RSA Public / Private keys TLS ECDSA Public / Private keys	CO User	N/A	None
List Certificate	Display / log expiration data of installed certificates	N/A	N/A	CO User	N/A	None
List private keys	List private keys	N/A	N/A	CO User	N/A	None
Audit Management Services						
View System Audit Log	Display logs/files of configuration changes	N/A	N/A	CO User	N/A	None
Export Analytics Logs System	Export analytics logs system	N/A	N/A	CO	N/A	None
Tenant Services						
Create Tenant	Create tenant deployment	N/A	N/A	CO User	N/A	None
Tenant SSH establish connection	Connecting to tenant-console via SSH	N/A	N/A	User	N/A	None
Tenant SSH close connection	Closing the tenant-console SSH session	N/A	N/A	User	N/A	None
System Management Services						
Configure SSH access options	Enable / Disable SSH access, configure IP address allow list	N/A	N/A	CO User	N/A	None
Configure SSH user configuration	Update ssh/ authorized_keys file for user authentication	N/A	SSH ECDSA public key SSH ECDSA private key	CO	W	None
Reboot System	Restart cryptographic module	N/A	SSPs listed in Table 12	CO	Z	Module reboots
Secure Erase	Full system zeroization	N/A	SSPs listed in Table 12	CO	Z	Module end of life
SSH Services						
Establish SSH session	Key authentication	ECDSA with SHA2-256 / SHA2-384 curves P-256 / P-384	SSH ECDSA public key SSH ECDSA private key	CO User	W	SSH connection successful
	Password authentication	N/A	Password	CO User	W	SSH connection successful

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
	Key exchange	ECDSA KeyGen, CTR_DRBG	SSH EC Diffie-Hellman public key SSH EC Diffie-Hellman private key	CO User	G	SSH connection successful
			DRBG Seed		E	
			DRBG internal state (V and key values)		W, E	
		KAS-ECC-SSC	SSH EC Diffie-Hellman public key	CO User	W	SSH connection successful
			SSH EC Diffie-Hellman private key		E	
			SSH shared secret		G	
	Key derivation	[SP 800-135] SSH KDF	SSH shared secret	CO User	E	SSH connection successful
			derived SSH session key (AES, HMAC)		G	
Maintain SSH Session	Data encryption and decryption	AES-CBC AES-CTR	derived SSH Session key (AES)	CO User	E	SSH connection successful
	Data integrity (MAC): HMAC-with SHA-1/ SHA2-256	HMAC	derived SSH session key (HMAC)	CO User	E	SSH connection successful
Close SSH Session	Close SSH session	N/A	SSH EC Diffie-Hellman key-pair; SSH shared secret; derived SSH session key	CO User	Z	SSH connection closed
TLS Services						
Establish TLS Session	SigGen / SigVer	ECDSA / RSA	TLS RSA Public / Private keys TLS ECDSA Public / Private keys	CO User	W	Service Indicator: Approved
	Key exchange	ECDSA KeyGen, CTR_DRBG	TLS EC Diffie-Hellman public key TLS EC Diffie-Hellman private key	CO User	G	Service Indicator: Approved
			DRBG Seed		E	
			DRBG internal state (V and key values)		W, E	
		KAS-ECC-SSC	TLS EC Diffie-Hellman public key		W	
			TLS EC Diffie-Hellman private key		E	
			TLS pre-primary secret		G	
	Key derivation	[SP 800-135] TLS KDF	TLS pre-primary secret	CO User	E	Service Indicator: Approved
			TLS primary secret		G, E	
			TLS derived session keys (AES and HMAC or authentication cypher)		G	

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Maintain TLS Session	Data encryption, data authentication	AES-CBC with HMAC-SHA2-256 / SHA2-384 or AES-GCM	Derived TLS session keys (AES and HMAC or authentication cypher)	CO User	E	Service Indicator: Approved
Close TLS session	Close TLS session	N/A	TLS EC Diffie-Hellman private key; TLS EC Diffie-Hellman public key; TLS pre-primary secret; TLS primary secret; TLS derived session keys	CO User	Z	TLS connection closed
Other services						
Show version	Return the module name and version	N/A	N/A	CO User	N/A	None
Show license	Return license indication	N/A	N/A	CO User	N/A	None
Show status	Return the module status	N/A	N/A	CO User	N/A	None
Self- test	Execute integrity test; Execute the CASTs	All the algorithms listed in table section 10	N/A (key for self-tests are not SSPs)	CO User	N/A	None
Show tenant	Lists tenant information	N/A	N/A	CO User	N/A	None

Table 8 - Approved Services

Table 9 shows the non-Approved services, a description, the non-Approved algorithms that are accessed, the role and service indicator, where applicable.

Service	Description	Algorithms Accessed	Role	Indicator
Establish TLS session	Signature generation and verification	algorithms listed in Table 4 rows DSA, RSA, ECDSA, EdDSA digital signature	User/ CO	No indicator
	Key exchange	- TLS KDF using MD5, SHA-1, SHA2-224, SHA2-512, SHA3 - Diffie-Hellman - RSA Key wrapping with all keys - EC Diffie-Hellman using curves other than P-256 and P-384 - EC Diffie-Hellman using P-256 and P-384 with Static Unified and OnePassDh	User/ CO	No indicator
Maintain TLS session	Data encryption	AES-CCM, AES-CFB, AES-OFB, AES-XTS, AES-KW, DES, RC4, Triple-DES, SM2, SM4	User/ CO	No indicator
	Data authentication	HMAC-SHA2-224, HMAC-SHA2-512, AES-CMAC, Triple-DES	User/ CO	No indicator
Create TLS key	Key generation	RSA Key Generation with modulus sizes other than 2048, 3072 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	No indicator

Service	Description	Algorithms Accessed	Role	Indicator
Key derivation	Key derivation	SNMP KDF IKEv1 KDF IKEv2 KDF	User/CO	No indicator
Message digest	Message digest	SHA2-224 SHA2-512 SM3 MD5	User/CO	No indicator

Table 9 - Non-Approved Services

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

5.2 On-Demand Integrity Test

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

5.3 Executable Code

The executable code is defined by the firmware version 1.5.1. All code belonging to this firmware version is the executable code of the module.

6 Operational Environment

6.1 Applicability

The module operates in a non-modifiable operational environment provided by F5 with firmware version 1.5.1. Once the module is operational, it does not allow the loading of any additional firmware.

The module is a firmware validated at a Security Level 2 in Physical Security then the security area is N/A.

7 Physical Security

The module tested in the platforms listed in Table 2 is enclosed in a hard-metallic production grade enclosure that provides opacity and prevents visual inspection of the internals. 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 on the module's platform to operate in approved mode of operation

Physical Security Mechanism	Recommended Frequency of Inspection / Test	Inspection/Test Guidance Details
Production grade enclosure (SL1)	N/A	N/A
Opaque enclosure (SL2)	N/A	N/A
Tamper Evident Labels (SL2)	Once per month	The CO checks the quality of the tamper evident labels for any sign of removal, replacement, tearing. In the event that 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 10 - Physical Security Inspection Guidelines

7.1 Tamper Label Placement

The pictures below show the location of all tamper-evident labels for each platform. Label application instructions are provided in Section 11.2.1 of the Crypto-Officer guidance below. The tamper label placements are delineated with red circles.

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

Table 11 – Number of Tamper Evident Labels per hardware appliance

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.
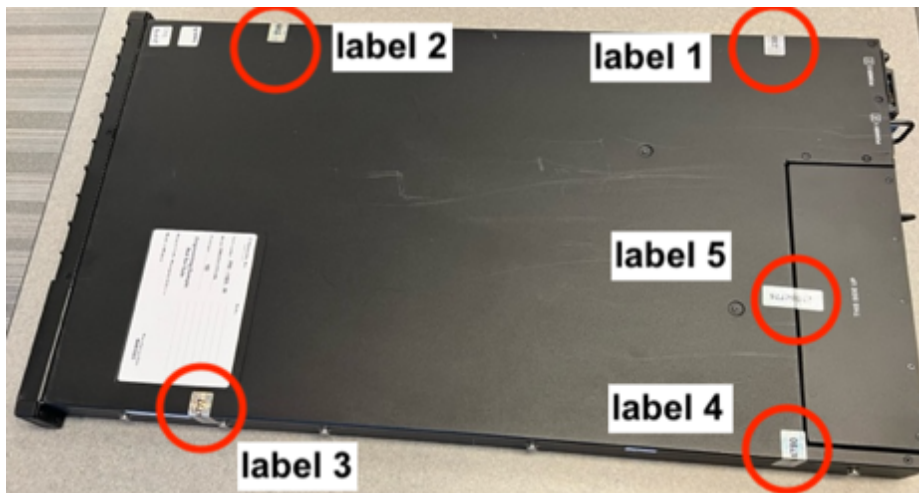

Figure 9 – Tamper labels on r10900, r10920-DF
(4 +1 tamper labels shown). Labels are located on the lateral sides of the platform -labels 1,2,3 and 4. The tamper label 5 on the chassis lid is covering the ventilation fan tray that allows access to SSD.

8 Non-invasive Security

This section is N/A until non-invasive security is defined.

9 Sensitive Security Parameter Management

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export[3]	Establi shment	Stor age	Zeroizatio n	Use and related SSPs
TLS RSA public key / asymm etric	112-bits and 150-bits	RSA A3896, A5260	Generated conformant to SP800-133r2 (CKG) using [FIPS 186-4] Key generation method; random values are obtained using [SP 800-90Ar1] DRBG	Public key input during protocol handshake Public key output during protocol handshake	N/A	SSD	Secure Erase	**Use**: Key generation, Digital signature verification used in the TLS protocol **Related SSPs**: TLS RSA private key, DRBG internal state (V and key values)
TLS RSA private key / asymm etric	112-bits and 150-bits	RSA A3896, A5260	Generated conformant to SP800-133r2 (CKG) using [FIPS 186-4] Key generation method; random values are obtained using [SP 800-90Ar1] DRBG	N/A	N/A	SSD	Secure Erase	**Use**: Key generation, Digital signature generation used in the TLS protocol **Related SSPs**: TLS RSA private key, DRBG internal state (V and key values)
TLS ECDSA public key / asymm etric	128-bits and 192-bits	ECDSA A3896, A5260	Generated conformant to SP800-133r2 (CKG) using [FIPS 186-4] ECDSA Key Generation method; random values are obtained using [SP 800-90Ar1] DRBG	Public key input during protocol handshake Public key output during protocol handshake	N/A	SSD	Secure Erase	**Use**: Key generation, Digital signature verification used in the TLS protocol **Related SSPs**: TLS ECDSA private key, DRBG internal state (V and key values)
TLS ECDSA private key /	128-bits and	ECDSA A3896, A5260	Generated conformant to SP800-133r2 (CKG) using	N/A	N/A	SSD	Secure Erase	**Use**: Key generation, Digital signature

---

[3] "CST Establishment" column defines the distribution and entry options from IG 9.5.A e.g. Automated Distribution / Electronic Entry = AD/EE

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export[3]	Establishment	Storage	Zeroization	Use and related SSPs
asymmetric	192-bits		[FIPS 186-4] ECDSA Key Generation method; random values are obtained using [SP 800-90Ar1] DRBG					generation used in the TLS protocol **Related SSPs**: TLS ECDSA public key, DRBG internal state (V and key values)
TLS EC Diffie-Hellman public key / asymmetric	128-bits and 192-bits	EC Diffie-Hellman A3896, A5260	Generated conformant to SP800-133r2 (CKG) using [FIPS 186-4] Key Generation; random values are obtained using [SP 800-90Ar1] DRBG	Public key input during protocol handshake Public key output during protocol handshake	N/A	RAM	Secure Erase; Closing TLS session; Reboot System	**Use**: Key generation, TLS protocol key exchange **Related SSPs**: TLS EC Diffie-Hellman private key, TLS pre-primary secret, DRBG internal state (V and key values)
TLS EC Diffie-Hellman private key / asymmetric	128-bits and 192-bits	EC Diffie-Hellman A3896, A5260	Generated conformant to SP800-133r2 (CKG) using [FIPS 186-4] Key Generation; random values are obtained using [SP 800-90Ar1] DRBG	N/A	N/A	RAM	Secure Erase; Closing TLS session; Reboot System	**Use**: Key generation, TLS protocol key exchange **Related SSPs**: TLS EC Diffie-Hellman public key DRBG, TLS pre-primary secret, DRBG internal state (V and key values)
TLS pre-primary secret	EC Diffie-Hellman: 128-bits and 192-bits	TLS KDF A3896, A5260	N/A	N/A	Established via SP800-56Ar3 during key agreement for EC Diffie-Hellman	RAM	Secure Erase; Closing TLS session; Reboot System	**Use**: TLS protocol **Related SSPs**: TLS EC Diffie-Hellman public key; TLS EC Diffie-Hellman private key; TLS primary secret

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export[3]	Establishment	Storage	Zeroization	Use and related SSPs
					cipher suites.			
TLS primary secret	256-bits	TLS KDF A3896, A5260	Derived from SP 800-135 TLS KDF	N/A	N/A	RAM	Secure Erase; Closing TLS session; Reboot System	**Use**: TLS protocol **Related SSPs**: TLS pre-primary secret; TLS derived key
TLS derived session key	128 and 256-bits (AES) 112 to 256-bits (HMAC)	AES HMAC A3896, A5260	Derived from SP 800-135 TLS KDF	N/A	N/A	RAM	Secure Erase; Closing TLS session; Reboot System	**Use**: TLS protocol **Related SSPs**: TLS pre-primary secret**,** TLS primary secret
SSH ECDSA public key / asymmetric	128 and 192-bits	ECDSA A3896, A5260	N/A	SSPs input during TLS/SSH sessions	N/A	SSD	Secure Erase	**Use**: SSH key-based authentication **Related SSPs**: SSH ECDSA private key
SSH ECDSA private key / asymmetric	128 and 192-bits	ECDSA A3896, A5260	N/A	N/A	N/A	SSD	Secure Erase	**Use**: SSH key-based authentication **Related SSPs**: SSH ECDSA public key
SSH EC Diffie-Hellman public key / asymmetric	128 and 192-bits	KAS-ECC-SSC A3896, A5260	Generated conformant to SP800-133r2 (CKG) using [FPIS 186-4] Key generation method; random values are obtained using [SP 800-90Ar1] DRBG	Public key output during protocol handshake Public key input during protocol handshake	N/A	RAM	Secure Erase; Closing SSH session or terminating the SSH application; Reboot System	**Use**: SSH handshake **Related SSPs**: SSH EC Diffie-Hellman private key, SSH shared secret, DRBG internal state
SSH EC Diffie-Hellman private key / asymmetric	128 and 192-bits	KAS-ECC-SSC A3896, A5260	Generated conformant to SP800-133r2 (CKG) using [FPIS 186-4] Key generation method; random values are obtained	N/A	N/A	RAM	Secure Erase; Closing SSH session or terminating the SSH applicatio	**Use**: SSH handshake **Related SSPs**: SSH EC Diffie-Hellman public key, SSH shared

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export[3]	Establishment	Storage	Zeroization	Use and related SSPs
			using [SP 800-90Ar1] DRBG				n; Reboot System	secret, DRBG internal state
SSH shared secret	128 and 256-bits	SSH KDF A3896, A5260	N/A	N/A	Established via SP800-56Ar3 KAS-ECC-SSC	RAM	Secure Erase; Closing SSH session or terminating the SSH application; Reboot System	**Use**: Key derivation; SSH shared secret; **Related SSPs**: SSH EC Diffie-Hellman public key, SSH EC Diffie-Hellman private key, SSH derived session key
SSH derived session key	128 and 256-bits AES) 112 and 256-bits (HMAC)	AES HMAC A3896, A5260	Derived from SP 800-135 SSH KDF	N/A	N/A	RAM	Secure Erase; Closing SSH session or terminating the SSH application; Reboot System	**Use**: Used in data encryption / decryption and MAC calculations in SSH protocol **Related SSPs**: SSH shared secret
Password	1/676,000,000 (see Table 7)	N/A	N/A	SSPs input during TLS/SSH sessions	N/A	SSD as hashed format	Secure Erase	**Use**: SSH authentication; WebUI login **Related SSPs**: N/A
Entropy input string	256 bits	ESV Cert. #E85	Obtained from non-physical entropy source	N/A	N/A	RAM	Secure Erase; Reboot System	**Use**: random number generation **Related SSPs**: DRBG seed
DRBG seed	256 bits	CTR_DRBG A3896, A5260	Derived from the entropy string as defined by [SP 800-90Ar1]	N/A	N/A	RAM	Secure Erase; Reboot System	**Use**: random number generation **Related SSPs**: Entropy input, DRBG internal state (V and key values)
DRBG internal state (V and	256 bits	CTR_DRBG A3896, A5260	Derived from the seed as defined by [SP 800-90Ar1]	N/A	N/A	RAM	Secure Erase; Reboot System	**Use**: random number generation **Related SSPs**: Entropy

Key/ SSP Name/ Type	Streng th	Security Function and Cert. Number	Generation	Import /Export[3]	Establi shment	Stor age	Zeroizatio n	Use and related SSPs
key values)								input, DRBG seed

Table 12 - SSPs

9.1 Random Bit Generation - Entropy Source

The module employs a Deterministic Random Bit Generator (DRBG) based on [SP800-90Ar1] 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 Table 13 to seed the DRBG.

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

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

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 ES is tested in the OEs listed in Table 2.

Entropy Source	Minimum number of bits of entropy	Details
ESV #E85 (non-physical noise source)	256-bits	The CPU Jitter RNG version 3.4.1 entropy source uses jitter variations caused by executing instructions and memory accessed. The entropy source has been shown to provide full 256-bits of entropy at the output of the SHA3-256 vetted conditioning function (#A3769).

Table 13 - Non-Deterministic Random Number Generation Specification

9.2 SSP Generation

The module implements RSA, ECDSA and EC Diffie-Hellman asymmetric key generation services compliant with [FIPS186-4], and using an [SP800-90Ar1] DRBG.

In accordance with FIPS 140-3 IG D.H, the cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per section 4 example 1 [SP800-133r2] (vendor affirmed).

The RSA and ECDSA key pairs used for Digital Signature Schemes are generated in accordance with section 5.1 of [SP800-133r2] and maps specifically to [FIPS 186-4].

The ECDH key pair used for Key Establishment are generated in accordance with section 5.2 of [SP800-133r2] i.e. key generation method specified in [SP 800-56Ar3]. For this module applicable method from [SP800-56Ar3] is 5.6.1.2 ECC Key Pair Generation which maps to [FIPS 186-4].

The module does not implement symmetric key generation as an explicit service. The HMAC and AES symmetric keys are derived from shared secrets by applying [SP 800-135]

as part of the TLS/ SSH protocols. The scenario maps to the [SP 800-133r2] section 6.2.1 *Symmetric keys generated using Key Agreement Scheme.*

9.3 SSP Establishment

The module provides the following key establishment services:

- EC Diffie-Hellman key agreement scheme compliant with SP800-56Ar3 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 [SP 800-135] TLS KDFs and [SP 800-135] SSH KDFs. EC Diffie-Hellman key agreement provides 128 or 192-bits of encryption strength.
- [SP 800-38F], IG D.G, key wrapping in the context of TLS protocol using an approved authenticated encryption mode (i.e. AES-GCM) provides 128 or 256 bits of encryption strength (AES Cert. #A3896, 5260)
- [SP 800-38F], IG D.G, key wrapping in the context of TLS protocol using a combination of approved AES encryption and HMAC authentication method provides 128 or 256 bits of encryption strength (AES and HMAC Certs. #A3896, #A5260).
- [SP 800-38F], IG D.G, key wrapping in the context of SSH protocol using a combination of approved AES-CBC or AES-CTR encryption mode and HMAC authentication method provides 128 or 256 bits of encryption strength (AES and HMAC Certs. #A3896, #A5260).

9.4 SSP Entry / Output

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 SP 800-135 TLS KDF.

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 SP 800-135 SSH KDF.

There are no encrypted SSPs that are directly entered.

9.5 SSP Storage

As shown in Table 12 the keys are stored in the volatile memory (RAM) in plaintext form and are destroyed when released by the appropriate zeroization calls or when the system is rebooted.

The static SSPs are persistently stored in plaintext in the module's non-volatile memory solid-state drive (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.

9.6 SSP Zeroization

The zeroization methods listed in Table 12, overwrites the memory occupied by keys with "zeros" or pre-defined values.

The zeroization of temporary values are performed when they are no longer needed.

The zeroization can be enforced by the crypto officer with the following services:

- Calling Reboot System service will clear the SSPs present in volatile memory RAM memory.
- For keys present in non-volatile memory, using Secure Erase service (which can only be triggered during reboot of the test platform) will perform a single pass zeroization erasing the entire module.

10 Self-tests

10.1 Pre-Operational Self-Tests

The pre-operational self-test are performed automatically when the module is powered on. At initialization the module performed pre-operational self-test (integrity test) and the conditional cryptographic algorithm tests (CASTs). Services are not available during the pre-operational self-test and the data output interface is inhibited. On successful completion of the pre-operational self-tests and CASTs, the module enters the approved mode and cryptographic services are available. If the module fails any of the tests, the module returns an error code, and transitions to an the error state where any cryptographic operations are prohibited.

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.

10.1.1   Pre-operational Software/Firmware Integrity Test

The integrity of the module is verified 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 (see Table 15). The HMAC-SHA-384 algorithm is self-tested prior to the integrity test being run.

10.2 Conditional Self-Tests

10.2.1   Conditional Cryptographic Algorithm Tests

The module performs cryptographic algorithm self-tests (CASTs) on all Approved cryptographic algorithms. The module performs the CASTs shown in Table 14 during power-up. The CASTs consist of Known Answer Tests for all the approved cryptographic algorithms, SP800-90B Health Tests for entropy source and SP800-90Ar1 Health Tests for DRBG.

Algorithm	Test
non-physical entropy source	SP800-90B health test (APT and RCT) classified as CAST: • at start-up: performed on 1,024 consecutive samples. • during runtime.
CTR_DRBG	CAST KAT with AES 256 bits with and without derivation function SP800-90Ar1 section 11.3 health tests
AES	CAST KAT of AES encryption / decryption separately with AES-GCM mode and 256-bit key CAST KAT of AES encryption / decryption separately with ECB mode and 128 bit-key
RSA	CAST KAT of RSA PKCS#1 v1.5 signature generation with 2048 bit key and SHA2-256 CAST KAT of RSA PKCS#1 v1.5 signature verification with 2048 bit key and SHA2-256
ECDSA	CAST KAT of ECDSA signature generation using P-256 and SHA2-256

Algorithm	Test
	CAST KAT of ECDSA signature verification using P-256 and SHA2-256
KAS-ECC-SSC	CAST KAT of shared secret computation with P-256 curve
HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384	CAST KAT of HMAC-SHA-1, CAST KAT of HMAC-SHA2-256 CAST KAT of HMAC-SHA2-384 (prior integrity tests during pre-operational self-tests)
SHA-1, SHA2-256, SHA2-384	CAST KATs for all SHA sizes are covered by the respective HMAC KATs (allowed per IG 10.3.B)
[SP800-135] KDF	SSH CAST KAT TLS1.2 CAST KAT

Table 14 – Conditional Cryptographic Algorithm Self-Tests

10.2.2   Conditional Pairwise Consistency Test

A pairwise consistency test is run whenever asymmetric keys (RSA, EC Diffie-Hellman or ECDSA) are generated. PCT for ECDSA and RSA Key Pair Generation used for digital signatures is tested by the calculation and verification of a digital signature. PCT for Key Pair Generation EC Diffie-Hellman is covered by EC Diffie-Hellman PCT tested by the calculation and verification of a digital signature (IG 10.3.A).

10.2.3   On-Demand Self-Test

On demand self-tests are performed by rebooting the module. This service performs the same cryptographic algorithm tests executed during pre-operational self-test and CASTs. During the execution of the periodic and on-demand self-tests, crypto services are not available, and no data output or input is possible. The PCTs are executed on demand during the key generation functions invocation.

10.3 Error States

In the error state, any data output or cryptographic operations are prohibited. The module must reboot or be re-loaded with a fresh image to clear the error condition.

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

Error State	Cause of Error	Status Indicator
Error State	HMAC-SHA2-384 integrity test failure	Module will not load
	Failure of any of the CAST	Module will not load
	Failure of any of the PCTs	Module will reboot
	Failure of the APT, RCT at runtime	Module will reboot
	Failure of the APT, RCT at restart (power on)	Module will not load

Table 15 - Error States

11 Life-cycle assurance

11.1 Delivery and Operation

The hardware platforms 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.
- Inspect the packaging for tampering or other issues.
- Ensure that the external labels match the expected delivery and the shipped product.
- Ensure that the components in the box match those on the documentation shipped with the product.
- Verify the hardware platform with the model number given on the shipping label and marked on the hardware platform itself.

11.2 Crypto Officer Guidance

The Crypto Officer should 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.

11.2.1   Installing Tamper Evident Labels

Before the hardware platform is installed in the production environment, tamper-evident labels must be installed in the location identified for each module in Section 7.1. The following steps should be taken when installing or replacing the tamper evident labels on the test platforms on which the module runs. The instructions are also included in *F5 Platforms: FIPS Kit Installation* provided with each 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.

11.2.2   Installing F5OS

Follow the instructions in the "*Initial Configuration*" guide for the initial setup and configuration of the firmware 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 of operation 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.

11.2.2.1   Version Confirmation

The Crypto Officer should call the show version service (with command "show system security fips-module" and "show system image"), then confirm that the provided version matches the validated module name and version (F5OS-A Cryptographic Module and Version 1.5.1). Any firmware loaded into the module other than version 1.5.1 is out of the scope of this validation and will mean that the module is not operating as a FIPS validated module.

11.2.2.2   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".

11.2.3   Additional 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 of the test platform on which the module executes should be disabled after the initial power on and communications setup of the hardware.

11.3 User 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. The Approved and non-Approved modes of operation are specified in section 2.3. The algorithm-specific information is listed in sub-section below.

11.3.1   AES GCM IV

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 [SP800-52r2] section 3.3.1.

11.3.2   RSA SigGen/SigVer

All the modulus sizes supported by the module have been ACVP tested (per IG C.F).

11.3.3   Legacy Algorithms

The use of SHA-1 within Digital Signature Verification is allowed for legacy use per SP800-131Ar2 section 9. This may only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M.

12 Mitigation of other attacks

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

Appendix A. Glossary and Abbreviations

ADC	Application Delivery Controller
AES	Advanced Encryption Standard
API	Application Programming Interface
ACVP	Automated Cryptographic Validation Protocol
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CLI	Command Line Interface
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
ECDSA	Elliptic Curve Digital Signature Algorithm
ESV	Entropy Source Validation
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
GMAC	Galois Message Authentication Code
HMAC	Hash Message Authentication Code
IKE	Internet Key Exchange
KAS	Key Agreement Schema
KAT	Known Answer Test
KDF	Key Derivation Function
KTS	Key Transport Scheme
KW	AES Key Wrap
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
PAA	Processor Algorithm Accelerators
PUD	Public Use Document
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SNMP	Simple Network Mail Protocol
SSC	Shared-Secret Computation
SSD	Solid State Drive
SSH	Secure Shell
SSP	Sensitive Security Parameter

TLS	Transport Layer Security
Triple-DES	Triple Data Encryption Standard
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

RFC 2313   **PKCS #1: RSA Encryption Version 1.5**
                 March 1998
                 https://datatracker.ietf.org/doc/html/rfc2313

RFC 5288   **AES Galois Counter Mode (GCM) Cipher Suites for TLS**
                 August 2008
                 https://www.ietf.org/rfc/rfc5288.txt

RFC 7627   **Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension**
                 September 2015
                 https://www.ietf.org/rfc/rfc7627.txt

SP800-38A   **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

SP800-38B   **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

SP800-38C   **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

SP800-38D   **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

SP800-38F   **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

SP800-52r2   **Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations**
August 2019
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf

SP800-56Ar3   **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

SP800-90Ar1   **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

SP800-90B   **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

SP800-131Ar2   **Transitioning the Use of Cryptographic Algorithms and Key Lengths**
March 2019
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf

SP800-133r2   **NIST Special Publication 800-133 Revision 2 - Recommendation for Cryptographic Key Generation**
June 2020
https://doi.org/10.6028/NIST.SP.800-133r2

SP800-135r1   **NIST Special Publication 800-135 Revision 1 - Recommendation for Existing Application-Specific Key Derivation Functions**
December 2011
http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf

SP800-140B   **NIST Special Publication 800-140B - CMVP Security Policy Requirements**
March 2020
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140B.pdf