F5, Inc.

Cryptographic Module for BIG-IP(R)

version 1.0.2u-fips

FIPS 140-3 Non-Proprietary Security Policy

document version 1.1

Last update: June 2024

Prepared by:

atsec information security corporation

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

www.atsec.com

Table of Contents

Copyrights and Trademarks

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

VMware ESXi™ is a registered trademark of VMware®, Inc.

Intel® Xeon® is a registered trademark of Intel® Corporation.

Dell is a registered trademark of Dell, Inc.

Azure and Hyper-V are registered trademarks of Microsoft

AWS is a trademark of Amazon.com, Inc.

1 General Information

This document is the non-proprietary FIPS 140-3 Security Policy for version 1.0.2u-fips of the Cryptographic Module for BIG-IP. It 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 an Overall Security Level 1 module.

ISO/IEC 24759 Section 6. [Number Below]	FIPS 140-3 Section Title	Security Level
1	General	1
2	Cryptographic Module Specification	1
3	Cryptographic Module Interfaces	1
4	Roles, Services, and Authentication	1
5	Software/Firmware Security	1
6	Operational Environment	1
7	Physical Security	Not Applicable
8	Non-invasive Security	Not Applicable
9	Sensitive Security Parameter Management	1
10	Self-Tests	1
11	Life-cycle Assurance	1
12	Mitigation of Other Attacks	Not Applicable

Table 1 - Security Levels

2 Cryptographic Module Specification

2.1 Module Overview, Embodiment Type

The Cryptographic Module for BIG-IP (hereafter referred to as “the module”) is a software library implementing general purpose cryptographic algorithms. The module is a multiple-chip standalone cryptographic module.

The software module provides cryptographic services to applications through an Application Program Interface (API). The module also interacts with the underlying operating system via system calls.

2.2 Operational Environments

The module has been tested on the following platforms with the corresponding module variants and configuration options with and without PAA:

#	Operating System	Hardware Platform	Processor	PAA/ Acceleration
1	BIG-IP 16.1.3.1 on VMware ESXi™ 6.5 hypervisor	Dell PowerEdge M620	Intel® Xeon® E5-2670 Sandy Bridge	AES-NI and SHA extensions
2	BIG-IP 16.1.3.1 on Hyper-V 10.0.20348.1 on Windows Server 2022	Dell PowerEdge R450	Intel® Xeon Silver 4309Y	AES-NI and SHA extensions
3	BIG-IP 16.1.3.1 on KVM on Ubuntu 20.04.2 LTS (Focal Fossa)	Dell PowerEdge M630	Intel® Xeon® E5-2690 v4 Broadwell	AES-NI and SHA extensions

Table 2 - Tested Operational Environments

In addition to the configurations tested by the atsec CST laboratory, vendor-affirmed testing was performed on the following platforms for the module by F5, Inc.

#	Operating System	Hardware Platform
1	BIG-IP 16.1.3.1 running on Microsoft Corporation Hyper-V Virtual Machine	Azure -cli 2.48.1 on Intel Xeon Platinum 8272CL processor
2	BIG-IP 16.1.3.1 running on Xen 4.2.amazon	AWS CLI 2.11.19 on Intel Xeon Scalable Processor – Cascade Lake 8259CL

Table 3 - Vendor Affirmed Operational Environments

Note: The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

2.3 Modes of Operations

The module supports two modes of operation:

- The Approved mode of operation where only approved or vendor affirmed functions can be used as specified in Table 4.
- The non-Approved mode of operation where only non-approved security functions can be used (Table 5).

The module becomes operational and enters the approved mode after pre-operational self-tests succeed. No operator intervention is required to reach this point. Once the module is operational, the mode of operation is implicitly assumed depending on the security function invoked and the security strength of the cryptographic keys. Using any non-approved algorithms from Table 5 will put the module in non-approved mode implicitly.

2.4 Security Functions

The table below lists all security functions of the module, including specific key size(s) employed for approved or vendor-affirmed security functions, and implemented modes of operation.

CAVP Cert[1]	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strengths (bits)	Use / Function
Assembler implementation				
A2762	AES [FIPS 197, SP800-38A, SP800 38C, SP800 38D]	ECB, CBC, CTR, GCM	128 / 192 / 256-bit AES key / strength from 128 to 256 bits	Encryption and decryption
A2762	AES [FIPS 197, SP800 38D]	GMAC	128 / 192 / 256-bit AES key / strength from 128 to 256 bits	MAC generation/ verification
A2762	KTS (AES) FIPS 197, SP800-38F]	GCM	128 / 256-bit AES key / strength from 128 and 256 bits	Key wrapping
A2762	Counter DRBG [SP800-90ARev1]	AES-256 in CTR mode, with/ without derivation function, prediction resistance enabled and disabled	DRBG seed, DRBG internal state (V and Key values) / strength is 256 bits	Random number generation
Vendor Affirmed	CKG [SP800-133Rev2]	RSA KeyGen	2048/ 3072/ 4096-bit modulus / strength from 112 to 150 bits	Key pair generation

[1] There are algorithms, modes, and key/moduli sizes that have been CAVP-tested but are not used by any approved service of the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in this table are used by an approved service of the module.

CAVP Cert[1]	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strengths (bits)	Use / Function
		ECDSA KeyGen	P-256, P- 384 / strength 128 and 192 bits	
		Safe Primes Key Generation	ffdhe2048, ffdhe3072, ffdhe4096 / strength from 112 to 150 bits	
A2762	RSA KeyGen [FIPS 186-4]	B.3.3 Probable primes with standard key format	2048/ 3072/ 4096-bit modulus size / strength from 112 to 150 bits	RSA key pair generation
A2762	RSA SigGen [FIPS 186-4]	PKCS 1.5 with SHA-256, SHA-384	2048/ 3072/ 4096-bit modulus/ strength from 112 to 150 bits	RSA signature generation
A2762	RSA SigVer [FIPS 186-4]	PKCS 1.5 with SHA-1, SHA2-256, SHA2-384	2048/ 3072/ 4096-bit modulus / strength from 112 to 150 bits	RSA signature verification
A2762	ECDSA KeyGen [FIPS 186-4]	Appendix B.4.2: Testing Candidates	ECDSA/ ECDH key pair P-256 and P-384 curves / strength 128 and 192 bits	ECDSA/ ECDH key pair generation
A2762	ECDSA KeyVer [FIPS 186-4]	N/A	ECDSA/ ECDH key pair with P-256 and P-384 curves / strength 128 and 192 bits	ECDSA/ ECDH public key verification
A2762	ECDSA SigGen [FIPS 186-4]	SHA2-256, SHA2-384	ECDSA P-256, P- 384 curves / strength 128 and 192 bits	ECDSA signature generation
A2762	ECDSA SigVer [FIPS 186-4]	SHA2-256, SHA2-384	ECDSA P-256, P- 384 curves / strength 128 and 192 bits	ECDSA signature verification
A2762	SHA [FIPS180-4]	SHA-1, SHA2-256, SHA2-384	N/A	Message digest
A2762	HMAC [FIPS 198]	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384	128-1024-bit HMAC key / strength from 112 to 256 bits	MAC generation/ verification
A2762	KAS-ECC-SSC [SP800-56ARev3]	Ephemeral Unified: KAS Role: initiator, responder	P-256, P-384 / strength 128 and 192 bits	EC Diffie-Hellman shared secret computation IG D.F scenario 2, path 1

CAVP Cert[1]	Algorithm and Standard	Mode / Method	Description / Key Size(s) / Key Strengths (bits)	Use / Function
A2762	Safe Primes key Generation / Verification	Safe prime	Safe Prime Groups: ffdhe2048, ffdhe3072, ffdhe4096 / strength from 112 to 150 bits	Safe primes key generation
A2762	KAS-FCC-SSC [SP800-56ARev3]	dhEphemeral: KAS Role: initiator, responder	ffdhe2048, ffdhe3072, ffdhe4096 / strength from 112 to 150 bits	Diffie-Hellman shared secret computation IG D.F scenario 2, path 1
AESNI-SSSE3 Implementation				
A2711	AES [FIPS 197, SP800-38A, SP800 38D]	ECB, CBC	128 / 192/ 256-bit AES key / strength from 128 to 256 bits	Encryption and decryption
A2711	AES [FIPS 197, SP800 38D]	GMAC	128 / 192/ 256-bit AES key / strength from 128 to 256 bits	MAC generation/ verification
A2711	AES [SP800-38F]	GCM	128 / 256-bit AES key / strength 128 and 256 bits	Key wrapping
A2711	Counter DRBG [SP800-90ARev1]	AES 256 in CTR mode, with derivation function, prediction resistance enabled	Entropy input string, seed, V and Key values / strength is 256 bits	Random number generation
A2711	SHA [FIPS180-4]	SHA-1	N/A	Message digest
A2711	HMAC [FIPS 198]	HMAC-SHA-1	128-1024-bit HMAC key / strength from 112 to 256 bits	MAC generation/ verification

Table 4 - Approved Algorithms

The module does not implement any non-Approved but Allowed algorithm in Approved mode of operation with no security claimed.

The module does not implement any Non-Approved Algorithms Allowed in the Approved Mode of Operation.

The table below lists Non-Approved security functions that are not Allowed in the Approved Mode of Operation.

Algorithm/Functions	Use/Function

AES with OFB, CCM, CFB, XTS, KW modes, Blowfish, Camellia, CAST5, DES, IDEA, RC2, RC4, SEED, SM2, SM4, Triple-DES	Encryption and decryption
SHA2-224, SHA2-512, SM3, MD4, MD5, MDC2, RIPEMD, Whirlpool	Message digest
HMAC-SHA2-224, HMAC-SHA2-512, AES CMAC, Triple-DES CMAC	MAC generation/ verification
RSA KeyGen with 1024 and greater than 4096 up to 16384 modulus	RSA key pair generation
RSA SigGen with PKCS #1 v1.5 scheme with modulus size 2048, 3072, 4096 bits with SHA-1, SHA2-224, SHA2-512	RSA signature generation
RSA SigGen with PKCS #1 v1.5 scheme with keys other than the ones listed in Table 4	
RSA SigGen with PSS, X9.31 schemes	
RSA SigVer with PKCS #1 v1.5 scheme with modulus size 2048, 3072, 4096 bits with SHA2-224, SHA2-512	RSA signature verification
RSA SigVer with PKCS #1 v1.5 scheme with keys other than the ones listed in Table 4	
RSA SigVer with PSS, X9.31 schemes	
ECDSA KeyGen with P-224, P-521 curves	ECDSA key pair generation
ECDSA KeyVer with P-224, P-521 curves	ECDSA public key verification
ECDSA SigGen with P-256, P-384 curves and SHA-1, SHA2-224, SHA2-512	ECDSA signature generation
ECDSA SigVer with P-256, P-384 curves with SHA2-224, SHA2-512	ECDSA signature verification
ECDSA with SM2	ECDSA signature generation
	ECDSA signature verification
RSA with modulus sizes up to 16384 bits	RSA encryption and decryption

DSA	Domain parameter generation
	Domain parameter verification
	DSA key pair generation
	DSA signature generation
	DSA signature verification
HMAC_DRBG and Hash_DRBG for all SHA sizes, CTR_DRBG with AES-128, AES-192, ANSI X9.31 RNG	Random number generation
Diffie-Hellman key agreement with groups other than ffdhe2048, ffdhe3072, ffdhe4096	Diffie-Hellman shared secret computation
EC Diffie-Hellman Ephemeral without KDF Unified with curves other than P-256, P-384	EC Diffie-Hellman shared secret computation
EC Diffie-Hellman without KDF one PassDh and StaticUnified	

Table 5 - Non-Approved Algorithms Not Allowed in the Approved Mode of Operation

2.5 Module Design and Components

The software block diagram below shows the module, its interfaces with the operational environment and the delimitation of its cryptographic boundary with red lines.

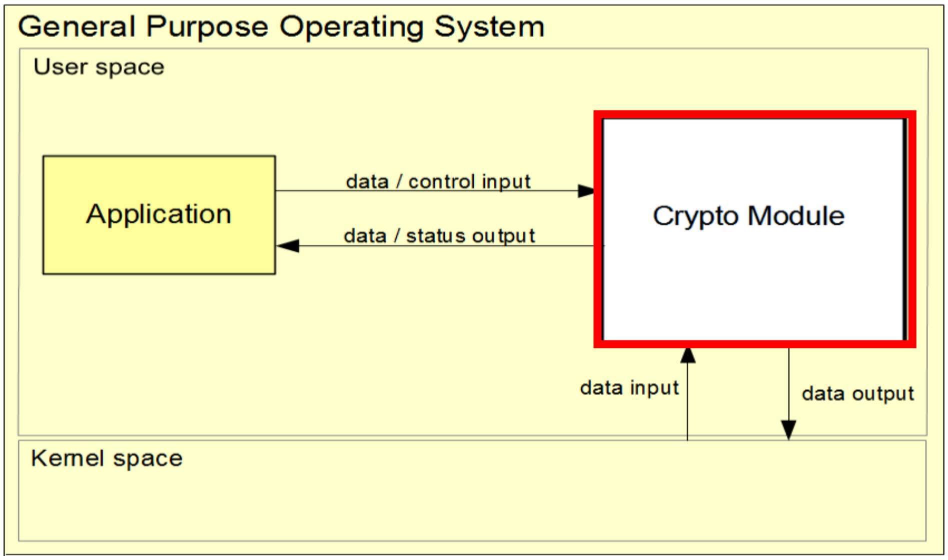

Figure 1 – Software Block Diagram

The module is implemented as a shared library. The cryptographic module boundary consists of two components:

- the shared library, the binary for cryptographic implementations (libcrypto.so.1.0.2u) and

- the file that holds the pre-computed integrity check value (.libcrypto.so.1.0.2u.hmac).

The module is aimed to run on a general-purpose computer; the physical perimeter is the surface of the case of the target platform, as shown with orange dotted lines in the diagram below:

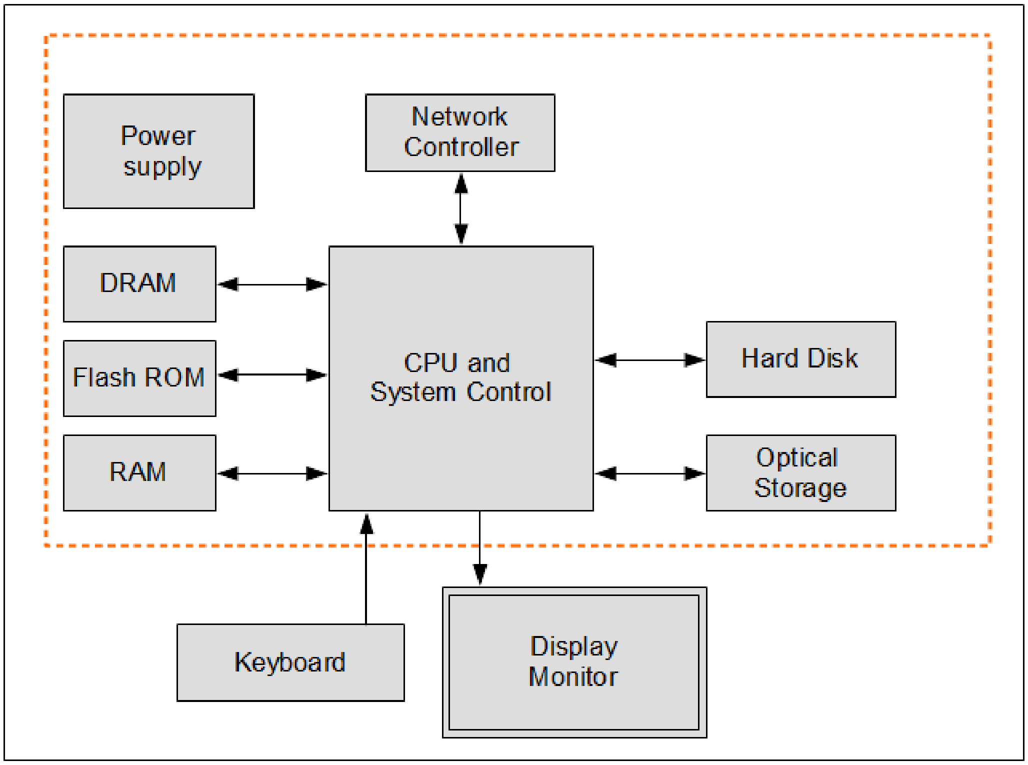

Figure 2 – Cryptographic Module Physical Perimeter

3 Cryptographic Module Ports and Interfaces

The logical interfaces are the API through which the applications request services. The following table summarizes the logical interfaces:

Physical Port	Logical Interface[2]	Data that passes over port/interface
As a software-only module, the module does not have physical ports. Physical Ports are interpreted to be the physical ports of the hardware platform on which it runs.	Data Input	API input parameters for data
	Data Output	API output parameters for data
	Control Input	API function calls for control
	Status Output	API return codes, error messages

Table 6 - Ports and Interfaces

Cryptographic bypass capability is not supported by the module. The module does not implement control output interface.

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

4 Roles, services, and authentication

The module supports the Crypto Officer role only. No support is provided for multiple concurrent operators or a Maintenance Operator.

4.1 Roles

Table below describes the authorized role(s) in which the service can be performed with specification of the service input parameters and associated service output parameters.

Role	Service	Input	Output
Crypto Officer	Encryption and decryption	Plaintext, key / ciphertext, key	Ciphertext / plaintext
	Key wrapping	Wrapping key, key to be wrapped / Unwrapping key, key and key to be unwrapped	Wrapped key / unwrapped key
	Random number generation	Number of bits	Random numbers
	RSA key pair generation	Key size	Public key, private key
	RSA signature generation	Private key, message, hashing algorithm	Computed signature
	RSA signature verification	Public key, digital signature, message, hashing algorithm	Pass/fail result of digital signature verification
	ECDSA/ ECDH key pair generation	Elliptic curve	Private key, public key
	ECDSA/ ECDH public key verification	Public key	Pass/fail result of public key verification
	ECDSA signature generation	Private key, message, hashing algorithm	Computed signature
	ECDSA signature verification	Public key, digital signature, message, hashing algorithm	Pass/fail result of digital signature verification
	EC Diffie-Hellman shared secret computation	Received public key, possessed private key	Shared secret
	Safe primes key generation	Group	Private key, public key
	Diffie-Hellman shared secret computation	Received public key, possessed private key	Shared secret
	Message digest	Message, hashing algorithm	Hashed message
	MAC generation	Message, key, MAC algorithm, MAC length	MAC tag
	MAC verification	MAC tag, key, MAC algorithm	Pass/fail result of MAC verification

	Show version	N/A	Name and version information
	Show status	N/A	Status output
	Self-tests	Power	Pass/fail results of self-tests
	Zeroization	Unencrypted SSPs listed in Table 10	Zeroized memory
	RSA encryption and decryption	Message, key	Ciphertext / plaintext
	Domain parameter generation	L and N pair	Domain parameters
	Domain parameter verification	Domain parameters	Pass/fail result of verification
	DSA key pair generation	Domain parameters	Public key, private key
	DSA signature generation	Private key, message, hashing algorithm	Computed signature
	DSA signature verification	Public key, digital signature, message, hashing algorithm	Pass/fail result of digital signature verification

Table 7 - Roles, Service Commands, Input and Output

4.2 Authentication

FIPS 140-3 does not require an authentication mechanism for level 1 modules. Therefore, the module does not implement an authentication mechanism for Crypto Officer. The Crypto Officer role is authorized to access all services provided by the module (see Table - Approved Services and Table - Non-Approved Services below).

4.3 Services

The table below lists all approved services that can be used in the approved mode of operation.

The status output from the FIPS_set_indicator_status service indicator's call is provided in Indicator column in Table 8. To read this indicator, the calling application must register a callback function using `FIPS_register_indicator_callback'. The callback function shall take the input of the form "char *" which is the form of the indicator being output by the module.

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Encryption and decryption	Executes AES-mode encrypt or decrypt operation	AES-ECB, AES-CBC, AES-CTR	AES key (128 / 192 / 256 bits)	Crypto Officer	W, E	AES-ECB, AES-CBC, AES-CTR

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Key wrapping	Executes AES-GCM key wrapping or unwrapping operation, per IG D.G	AES-GCM	AES key (128 / 256 bits)	Crypto Officer	W, E	AES-GCM
Random number generation	Generate random number	Counter DRBG	Entropy input string	Crypto Officer	W, E	CTR-DRBG-AES-256
			DRBG seed		G, E	
			DRBG internal state (V and Key values)		G, E	
RSA key pair generation	Generate RSA Key Pair	RSA KeyGen (FIPS 186-4) CKG [SP800-133Rev2], Counter DRBG	RSA private key, RSA public key (2048/ 3072/ 4096 bits)	Crypto Officer	G, R	RSA-KEY-GEN-2048, RSA-KEY-GEN-3072 RSA-KEY-GEN- 4096
RSA signature generation	Sign a message with a specified RSA private key	RSA SigGen (FIPS 186-4)	RSA private key (2048 / 3072 / 4096 bits)	Crypto Officer	E, W	RSA-SIG
RSA signature verification	Verify the signature of a message with a specified RSA public key	RSA SigVer (FIPS 186-4)	RSA public key (2048/ 3072 / 4096 bits)	Crypto Officer	E, W	RSA-VER
ECDSA/ ECDH key pair generation	Generate a keypair for a requested elliptic curve	ECDSA KeyGen (FIPS 186-4) CKG [SP800-133Rev2], Counter DRBG	ECDSA private key, ECDSA public key, EC Diffie-Hellman private key, EC Diffie-Hellman public key (P-256 and P-384 curves)	Crypto Officer	G, R	EC-KEYGEN-P-256, EC-KEYGEN-P-384
ECDSA/ ECDH public key verification	Public key verification	ECDSA KeyVer (FIPS 186-4)	ECDSA public key, EC Diffie-Hellman public key (P-256 and P-384 curves)	Crypto Officer	E, W	EC-KEY-VERIFY-P-256, EC-KEY-VERIFY-P-384
ECDSA signature generation	Sign a message with a specified ECDSA private key	ECDSA SigGen (FIPS 186-4)	ECDSA private key (P-256 and P-384 curves)	Crypto Officer	W, E	ECDSA-SIGN-P-256, ECDSA-SIGN-P-384

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
ECDSA signature verification	Verify the signature of a message with a specified ECDSA public key	ECDSA SigVer (FIPS 186-4)	ECDSA public key (P-256 and P-384 curves)	Crypto Officer	W, E	ECDSA-VERIFY-P-256, ECDSA-VERIFY-P-384
EC Diffie-Hellman shared secret computation IG D.F scenario 2, path 1	Calculate a shared secret via the ECDH algorithm	KAS-ECC-SSC Sp800-56Ar3	EC Diffie-Hellman private key (P-256 and P-384 curves)	Crypto Officer	W, E	ECDH-COMPUTE-KEY-P-256, ECDH-COMPUTE-KEY-P-384
			EC Diffie-Hellman shared secret		G, R	
			EC Diffie-Hellman public key (remote peer public key) (P-256 and P-384 curves)		W, E	
Safe primes key generation	Generate a keypair / verify public key	Safe Primes Key Generation, Safe Primes Key Verification	Diffie-Hellman private key (ffdhe2048, ffdhe3072, ffdhe4096)	Crypto Officer	G, R	FFDHE2048-KEYGEN, FFDHE3072-KEYGEN, FFDHE4096-KEYGEN
			Diffie-Hellman public key (ffdhe2048, ffdhe3072, ffdhe4096)		G, R, W, E	
Diffie-Hellman shared secret computation IG D.F scenario 2, path 1	Calculate a shared secret via the DH algorithm.	KAS-FFC-SSC Sp800-56Ar3	Diffie-Hellman private key (ffdhe2048, ffdhe3072, ffdhe4096)	Crypto Officer	W, E	FFDHE2048-COMPUTE, FFDHE3072- COMPUTE, FFDHE4096- COMPUTE
			Diffie-Hellman shared secret		G, R	
			Diffie-Hellman public key (remote peer public key) (ffdhe2048, ffdhe3072, ffdhe4096)		W, E	

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Message digest	Generate a digest for the requested algorithm	SHA-1, SHA2-256, SHA2-384	N/A	Crypto Officer	N/A	MESSAGE-DIGEST-SHA-1/ SHA-256/SHA-384
MAC generation/ verification	Generate/ Verify an HMAC or GMAC digest using the requested SHA algorithm or AES algorithm as appropriate	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, AES-GMAC	HMAC key, AES key	Crypto Officer	W, E	MSG-AUTH-HMAC-SHA-1, MSG-AUTH-HMAC-SHA-256 MSG-AUTH-HMAC-SHA-384 AES-GMAC
Show version	Return the SW version and the module's name	N/A	N/A	Crypto Officer	N/A	None
Show status	Return the module status	N/A	N/A	Crypto Officer	N/A	None
Self-tests	Execute self-tests	AES-ECB, AES-GCM, HMAC-SHA1, HMAC-SHA2-256, HMAC-SHA2-384, RSA (SigGen (FIPS 186-4), RSA SigVer (FIPS 186-4), KAS-ECC-SSC Sp800-56Ar3, KAS-FFC-SSC Sp800-56Ar3, ECDSA SigGen (FIPS 186-4) / ECDSA SigVer (FIPS 186-4), Counter DRBG	N/A (key for self-tests are not SSPs)	Crypto Officer	N/A	None
Zeroization	Zeroize all non-protected SSPs	n/a	All SSPs	Crypto Officer	Z	None

Table 8 - Approved Services

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.

The table below lists all non-Approved services that can only be used in the non-Approved mode of operation.

Service	Description	Algorithms Accessed	Role	Indicator
Encryption and decryption	Encryption/ decryption	AES with OFB, CFB, CCM, XTS, KW modes Triple-DES Blowfish, Camellia, CAST5, DES, IDEA, RC2, RC4, SEED, SM2, SM4	Crypto Officer	None
Message digest	Generating message digest	SHA2-224, SHA2-512, SM3, MD4, MD5, MDC2, RIPEMD, Whirlpool		None
MAC generation/ verification	MAC computation	HMAC-SHA2-224, HMAC-SHA2-512 AES CMAC, Triple-DES CMAC		None
RSA key pair generation	Generating key pair	RSA KeyGen with 1024, greater than 4096 and up to 16384 modulus		None
RSA signature generation	Generating signature	RSA SigGen with PKCS #1 v1.5 with keys other than the one listed in Table 4		None
		RSA SigGen with PSS, X9.31 schemes		
		RSA SigGen PKCS #1 v1.5 scheme with modulus size 2048, 3072, 4096 bits with SHA-1, SHA2-224, SHA2-512		
RSA signature verification	Verifying signature	RSA SigVer with PKCS #1 v1.5 with keys other than the one listed in Table 4		
		RSA SigVer with PKCS #1 v1.5 scheme with modulus size 2048, 3072, 4096 bits with SHA2-224, SHA2-512		
		RSA SigVer with PSS, X9.31 schemes		
ECDSA key pair generation	Generating key pair	ECDSA KeyGen using P-224, P-521 curves		None
ECDSA public key verification	Verifying public key	ECDSA KeyVer using P-224, P-521 curves		
ECDSA signature generation	Generating signature	ECDSA SigGen with P-256 and P-384 curves with SHA-1, SHA2-224 and SHA2-512; ECDSA with SM2		None
ECDSA signature verification	Verifying signature	ECDSA SigVer with P-256 and P-384 curves with SHA2-224 and SHA2-512; ECDSA with SM2		

Service	Description	Algorithms Accessed	Role	Indicator
RSA encryption and decryption	Encryption/ decryption	RSA with modulus sizes up to 16384 bits		None
Domain parameter generation	Generating domain parameters	DSA		None
Domain parameter verification	Verifying domain parameters			
DSA key pair generation	Generating key pair			
DSA signature generation	Generating signature			
DSA signature verification	Verifying signature			
Random number generation	Generating deterministic random number	HMAC_DRBG and Hash_DRBG for all SHA sizes		None
		CTR_DRBG with AES-128 or AES-192		None
		ANSI X9.31 RNG		None
Diffie-Hellman shared secret computation	Calculating shared secret	Diffie-Hellman key agreement with groups other than ffdhe2048, ffdhe3072, ffdhe4096		None
EC Diffie-Hellman shared secret computation		EC Diffie-Hellman Ephemeral without KDF Unified with curves other than P-256, P-384		
		EC Diffie-Hellman without KDF one PassDh and StaticUnified		

Table 9 - Non-Approved Services

5 Software/Firmware security

5.1 Integrity Techniques

The integrity of the module is verified by comparing a HMAC value calculated at run time on the libcrypto.so.1.0.2u file, with the HMAC-SHA2-256 value stored in the module file .libcrypto.so.1.0.2u.hmac that was computed at build time.

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

5.2 On-Demand Integrity Test

The on-demand integrity test is performed as part of the Pre-Operational Self-Tests by power-cycling the module.

6 Operational Environment

6.1 Applicability

The module operates in a modifiable operational environment. The module runs on a BIG-IP 16.1.3.1 operating system executing on the hardware and hypervisor specified in section 2.2. BIG-IP consists of a Linux based operating system customized for performance that runs directly on the hardware or in virtual environment.

6.2 Requirements

The module should be installed as stated in section 11. The operator should confirm that the module is installed correctly by sub-section 11.2.

7 Physical Security

The module is comprised of software only and therefore this section is Not Applicable (N/A).

8 Non-invasive Security

Currently the non-invasive Security is not required by FIPS 140-3 (see NIST SP 800-140F).

9 Sensitive Security Parameters Management

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroization	Use and related keys
AES key / CSP/ symmetric	128 to 256 bits	ECB, CBC, CTR: A2762 ECB, CBC: A2711	N/A	Input as an API parameter No export	N/A	RAM	EVP_CIPHER_CTX_cleanup	Use: Encryption and decryption; Related keys: N/A
AES key / CSP/ symmetric	128 to 256 bits	GMAC: A2762, A2711	N/A	Input as an API parameter No export	N/A	RAM	EVP_CIPHER_CTX_cleanup	Use: MAC generation/ verification; Related keys: N/A
AES key / CSP/ symmetric	128 and 256 bits	AES-GCM: A2762, A2711	N/A	Input as an API parameter No export	N/A	RAM	FIPS_cipher_ctx_cleanup()	Use: Key wrapping; Related keys: N/A
HMAC key / CSP/ symmetric	112 to 256 bits	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384: A2762 HMAC-SHA-1: A2711	N/A	Input as an API parameter No export	N/A	RAM	HMAC_CTX_cleanup()	Use: MAC generation/ verification; Related keys: N/A
RSA private key / CSP/ asymmetric	112 to 150 bits	RSA SigGen: A2762	Generated conformant to section 5.1 of SP800-133Rev2 (CKG) using [FPIS 186-4], Appendix B.3.3 key generation method and the random	Import/ Export: CM to/ from TOEPP Path. Passed to/ from the module via API paramete	N/A	RAM	FIPS_rsa_free()	Use: RSA key pair generation, digital signature generation; Related keys: RSA public key, DRBG internal state (V

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroization	Use and related keys
			value used in the key generation is obtained using [SP 800-90ARev1] DRBG	rs in plaintext format.				and Key values)
RSA public key / PSP/ asymmetric		RSA SigVer: A2762						Use: RSA key pair generation, digital signature verification; Related keys: RSA private key, DRBG internal state (V and Key values)
ECDSA private key / CSP/ asymmetric	128 and 192 bits	ECDSA SigGen: A2762	Generated conformant to section 5.1 of SP800-133Rev2 (CKG) using [FPIS 186-4], Appendix B.4.2 key generation method and the random value used in the key generation is obtained using [SP 800-90ARev1] DRBG	Import/ Export: CM to/ from TOEPP Path. Passed to/ from the module via API parameters in plaintext format.	N/A	RAM	EC_KEY_free()	Use: ECDSA/ ECDH key pair generation, digital signature generation; Related keys: ECDSA public key, DRBG internal state (V and Key values)
ECDSA public key / PSP/ asymmetric		ECDSA SigVer: A2762						Use: ECDSA/ ECDH key pair generation, digital signature verification; Related keys:

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroization	Use and related keys
								ECDSA private key, DRBG internal state (V and Key values)

EC Diffie-Hellman private key / CSP/ asymmetric	128 and 192 bits	KAS-ECC-SSC Sp800-56Ar3: A2762	Generated conformant to section 5.2 of SP800-133Rev2 (CKG) using [FIPS 186-4], Appendix B.4.2 key generation method and the random value used in the key generation is obtained using [SP800-90ARev1] DRBG	Import/ Export: CM to/ from TOEPP Path.
				Passed to/ from the module via API parameters in plaintext format.
					N/A	RAM	EC_KEY_free()
							EC_POINT_free()
								Use: EC Diffie-Hellman shared secret computation;
								Related keys: EC Diffie-Hellman public key, DRBG internal state (V and Key values), EC Diffie-Hellman shared secret

EC Diffie-Hellman public key / PSP/ asymmetric								Use: EC Diffie-Hellman shared secret computation; Related keys: EC Diffie-Hellman private key, DRBG internal state (V and Key values), EC Diffie-Hellman

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroization	Use and related keys
								shared secret

EC Diffie-Hellman shared secret / CSP/ asymmetric	128 and 192 bits	KAS-ECC-SSC Sp800-56Ar3: A2762	N/A	No import
				Export: CM to TOEPP Path.
				Passed from the module via API parameters in plaintext format.
					Established via SP800-56ARev3 KAS-ECC-SSC	RAM	EC_KEY_free()
							EC_POINT_free()
								Use: EC Diffie-Hellman shared secret computation;
								Related keys: EC Diffie-Hellman private key, EC Diffie-Hellman public key

Diffie-Hellman private key / CSP/ asymmetric	112 to 150 bits	KAS-FFC-SSC Sp800-56Ar3: A2762	Generated conformant to section 5.2 of SP800-133Rev2 (CKG) using [SP800-56Ar3], Section 5.6.1.1.4 key generation method and the random value used in the key generation is obtained using [SP800-90ARev1] DRBG	Import/ Export: CM to/ from TOEPP Path. Passed to/ from the module via API parameters in plaintext format.	N/A	RAM	DH_free	Use: Diffie-Hellman shared secret computation; Related keys: Diffie-Hellman public key, DRBG internal state (V and Key values)
Diffie-Hellman public key / PSP/ asymmetric								Use: Diffie-Hellman shared secret computation; Related keys: Diffie-Hellman private

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroization	Use and related keys
								key, DRBG internal state (V and Key values)
Diffie-Hellman shared secret / CSP/ asymmetric	112 to 150 bits	KAS-FFC-SSC Sp800-56Ar3: A2762	N/A	No import Export: CM to TOEPP Path. Passed from the module via API parameters in plaintext format.	Established via SP800-56ARev3 KAS-FFC-SSC	RAM	DH_free	Use: Diffie-Hellman shared secret computation; Related keys: Diffie-Hellman private key, Diffie-Hellman public key
Entropy input string (IG D.L) /CSP	256 bits	Counter DRBG: A2762, A2711, ESV: E16	Generated by the entropy source (ESV Cert. #E16) (reference in section 11.2)	Import from the OS No Export	N/A	RAM	when the system is powered down	Use: Random number generation; Related keys: DRBG seed
DRBG seed (IG D.L), /CSP	256 bits	Counter DRBG: A2762, A2711	Derived from the entropy input string as defined by [SP 800-90ARev1]	No import: it remains within the cryptographic boundary. No Export	N/A	RAM	FIPS_drbg_uninstantiate	Use: Random number generation; Related keys: Entropy input string, DRBG Internal state (V and Key values)
DRBG internal state (V and Key values)	256 bits	Counter DRBG: A2762, A2711	Derived from the seed as defined by [SP 800-90ARev1]	No import: it remains within the cryptogra	N/A	RAM	FIPS_drbg_uninstantiate	Use: Random number generation;

Key/ SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import /Export	Establishment	Storage	Zeroization	Use and related keys
(IG D.L) /CSP				phic boundary No Export				Related keys: DRBG seed (V and Key values), RSA private key, RSA public key, ECDSA private key, ECDSA public key, EC Diffie-Hellman private key, EC Diffie-Hellman public key, Diffie-Hellman private key, Diffie-Hellman public key

Table 10 - SSPs

9.1 Random bit Generator

The module employs a Deterministic Random Bit Generator (DRBG) based on [SP800-90ARev1] for the generation of random value used in asymmetric keys, and for providing a RNG service to calling applications. The Approved DRBG provided by the module is the Counter DRBG with AES-256. The module uses the Entropy source specified in Table 11 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 ES is tested in the OEs listed in Table 1.

Entropy Source	Minimum number of bits of entropy	Details
ESV #E16 (non-physical noise source)	256	CPU Jitter 3.4.0 entropy source with SHA-3 as the vetted conditioning component is located within the physical perimeter of the module but outside the cryptographic boundary of the module.

Table 11 - Non-Deterministic Random Number Generation Specification

9.2 SSP generation

The module generates SSPs in accordance with FIPS 140-3 IG D.H. The cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per section 4 [SP800-133Rev2] (vendor affirmed), using DRBG compliant with [SP800-90ARev1]. A seed (i.e., the random value) used in asymmetric key generation is a direct output from [SP800-90ARev1] Counter DRBG. The following methods are implemented:

- RSA KeyGen (FIPS186-4), according to Appendix B.3.3 of FIPS 186-4, compliant with SP800-133r2, Section 5.1: generates 2048, 3072 and 4096-bit keys with 112-150 bits of strength.
- ECDSA KeyGen (FIPS186-4), according to Appendix B.4.2 of FIPS 186-4, compliant with SP800-133r2, Section 5.1 and Section 5.2: P-256 and P-384 curves with 128 and 192 bits of key strength. Note that this generation method is also used to generate ECDH key pairs.
- Safe Primes Key Generation (SP80056Ar3): according to Section 5.6.1.1 of SP800-56Ar3, compliant with SP800-133r2, Section 5.2: Safe Prime Groups: ffdhe2048, ffdhe3072, ffdhe4096 with 112-150 bits of strength.

The key generation services for RSA, Diffie-Hellman, EC Diffie-Hellman and ECDSA key pairs as well as the [SP 800-90ARev1] DRBG have been CAVP tested with certificates found in Table 4.

The module does not implement symmetric key generation.

9.3 SSP entry and output

The module does not support manual SSP entry or intermediate key generation output. The module does not support entry and output of SSPs beyond the physical perimeter of the operational environment. The SSPs can be provided to the module in plaintext form via API parameters, to and from the calling application running on the same operational environment. This is allowed by [FIPS 140-3_IG] IG 9.5.A Table 1, according to the “CM Software to/from App via TOEPP Path” entry which refers to keys communicated within the physical perimeter of the GPC.

9.4 SSP establishment

The module provides:

- KAS-FFC-SSC, SP800-56Ar3

The module implements Diffie-Hellman shared secret computation, compliant with SP800-56ARev3 and scenario 2 (path 1) only of IG D.F. The shared secret computation with groups ffdhe2048, ffdhe3072, ffdhe4096, SSP establishment methodology, provides between 112 and 150 bits of strength.

- KAS-ECC-SSC, SP800-56Ar3

The module implements EC Diffie-Hellman shared secret computation, compliant with SP800-56ARev3 and scenario 2 (path 1) only of IG D.F. The shared secret computation with curves P-256 or P-384, SSP establishment methodology, provides 128 or 192 bits of strength.

- AES-GCM Key Wrapping (KTS)

The module also provides key wrapping is used in the context of using TLS protocol implemented outside of the module boundary to send and receive key material in the payload. The key wrapping methods are provided by the TLS record layer using an approved authenticated encryption mode (i.e. AES-GCM). The key wrapping method using AES-GCM is an approved key transport method according to IG D.G. AES-GCM, SSP establishment methodology, provides 128 or 256 bits of strength. The TLS protocol has not been reviewed or tested by the CAVP or CMVP.

9.5 SSP storage

SSPs are provided to the module by the calling process and are destroyed when released by the appropriate API function calls.

The module does not perform persistent storage of SSPs.

9.6 SSP Zeroization

The memory occupied by SSPs and keys is allocated by regular memory allocation operating system calls. The application is responsible for calling the appropriate destruction functions provided in the module's API. The destruction functions (listed in Table 10) overwrite the memory occupied by keys with “zeros” and deallocate the memory with the regular memory deallocation operating system call.

10 Self-tests

10.1 Pre-operational Tests

Pre-operational self-tests are performed automatically when the module is loaded into memory; the pre-operational self-tests ensure that the module is not corrupted and that the cryptographic algorithms work as expected.

While the module is executing the pre-operational self-tests, services are not available, and input and output are inhibited. The module does not return control to the calling application until the tests are completed. On successful completion of the pre-operational self-tests, the module enters operational mode and cryptographic services are available. If the module fails any of the tests, it will return an error code and enter the error state to prohibit any further cryptographic operations.

The module provides the Self-Test service to perform periodic and on-demand self-tests. Both periodic and on demand self-tests (i.e., Conditional Cryptographic Algorithm Self-Tests (CASTs) and integrity test) can be invoked by powering-off and reloading 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.

10.1.1 Pre-operational Software Integrity Test

The integrity of the module is verified by comparing an HMAC-SHA2-256 value calculated at runtime with the HMAC-SHA2-256 value stored in the module that was computed at build time.

Prior to using HMAC-SHA2-256, a CAST is performed. If the CAST on the HMAC-SHA2-256 is successful, the HMAC value of the runtime image is recalculated and compared with the stored HMAC value pre-computed at compilation time.

10.2 Conditional Self-Tests

The following sub-sections describe the conditional self-tests supported by the module. If one of the Conditional self-tests fails, the module transitions to the ‘Halt Error’ state and a corresponding error indication is given. While the module is executing the CASTs, services are not available, and input and output are inhibited.

The entropy source performs its required self-tests; those are not listed in this section, as the entropy source is not part of the cryptographic boundary of the module.

10.2.1 Conditional Cryptographic algorithm tests

The module performs cryptographic algorithm self-tests (CASTs) on all Approved cryptographic algorithms. The module performs CASTs before the integrity test. The CASTs consist in Known Answer Tests for all the approved cryptographic algorithms and the SP800-90ARev1 Health Tests for DRBG.

Algorithm	Test
Counter DRBG	KAT with AES 256 bits with derivation function SP800-90ARev1 section 11.3 health tests
AES-ECB	Encryption KAT with 128 bit-key
	Decryption KAT with 128 bit-key

Algorithm	Test
AES-GCM	Encryption KAT with 128-bit key
	Decryption KAT with 128-bit key
RSA	PKCS#1 v1.5 signature generation KAT with 2048 bit key and SHA2-256
	PKCS#1 v1.5 signature verification KAT, with 2048 bit key and SHA2-256
ECDSA	Signature generation KAT, with P-256 and SHA2-256
	Signature verification KAT, with P-256 and SHA2-256
KAS-ECC-SSC	"Z" computation KAT with P-256 curve
KAS-FFC-SSC	"Z" computation KAT with 2048 modulus
HMAC-SHA	HMAC-SHA-1 KAT
	HMAC-SHA2-256 KAT
	HMAC-SHA2-384 KAT
SHA	KATs for all SHA sizes are covered by respective HMAC KATs (allowed per IG 10.3.B)

Table 12 - Conditional Cryptographic Algorithm Self-Tests

10.2.2 Conditional Pairwise Consistency Test

A pairwise consistency test (PCT) is run whenever asymmetric keys (RSA, DH, ECDH/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 Diffie-Hellman key pair generation is performed following the SP 800-56Arev3 requirements. PCT for EC Diffie-Hellman key pair generation is covered by ECDSA PCT (IG 10.3.A). While the module is executing the PCTs, services are not available, and input and output are inhibited.

10.3 Error States

Error State	Cause of Error	Status Indicator
Halt Error The module must be re-loaded in order to clear the error condition. That data output is inhibited.	HMAC-SHA2-256 KAT failure or HMAC-SHA2-256 integrity test failure	Module will not load
	Failure of any of the CASTs	Error message related to the crypto function listed in Table 12 and the flag 'fips_selftest_fail' is set.

Error State	Cause of Error	Status Indicator
	Failure of any of the PCTs	Error message a PCT failure for RSA, DH, ECDH or ECDSA pairwise consistency test and the flag 'fips_selftest_fail' is set.

Table 13 - Error States

11 Life-cycle assurance

11.1 Delivery and Operation

The module i.e. 1.0.2u-fips binary and its integrity check file are distributed and installed as a part of the BIG-IP product ISO.

There are no maintenance requirements.

11.2 Crypto Officer Guidance

The FIPS validated module activation requires installation of the BIG-IP System License key file. The Crypto Officer should install this file as /config/bigip.license and verify the FIPS validated module license activation (or reactivation) by running the command: ***'tmsh show sys license***' which should output FIPS 140, BIG-IP VE-1G to 10G,' under the 'Active Modules' list. After the FIPS validated module license is installed, the command prompt will change to 'REBOOT REQUIRED'. The Crypto Officer must reboot the BIG-IP for all FIPS-compliant changes to take effect.

On the BIG-IP product the Crypto Officer should call the dedicated Show version API, fips_get_f5fips_module_version, to ensure that the module identifier and version are shown as: Cryptographic Module for BIG-IP OpenSSL 1.0.2u-fips 20 Dec 2019.

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/16

11.2.1 AES GCM IV

The Crypto Officer shall consider the following requirements and restrictions when using the module.

> For TLS 1.2, the module offers the AES-GCM implementation and uses the context of Scenario 1 of IG C.H. The module is compliant with SP800-52Rev2 section 3.3.1 and the mechanism for IV generation is compliant with RFC5288.
>
> The module does not implement the TLS protocol. The module's implementation of AES-GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS protocol implicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key.
>
> In the event the module's power is lost and restored, the consuming application must ensure that a new key for use with the AES-GCM key encryption or decryption under this scenario shall be established.

11.2.2 SP800-56Ar3 Assurances

To comply with the assurances found in Section 5.6.2 of SP 800-56Ar3, the keys for KAS-FFC-SSC and KAS-ECC-SSC must be generated using the approved key generation services specified in section 9.2. For KAS-FFC-SSC the module generates keys using Safe Primes Key Generation with Safe Prime Groups: ffdhe2048, ffdhe3072, ffdhe4096. For KAS-ECC-SSC, the module generates keys using ECDSA KeyGen, Testing Candidates, with curves P-384 and P-256. The module performs full public key validation on the generated public keys. Additionally, the module performs full public key validation on the received public keys.

11.2.3 RSA Digital Signature

Per IG C.F, the module implements FIPS 186-4 RSA SigVer and RSA SigGen with modulus lengths of 2048, 3072, 4096 bits. All these modulus lengths have been CAVP tested.

12 Mitigation of other attacks

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

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ESV	Entropy Source Validation
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
KAS	Key Agreement Schema
KAT	Known Answer Test
KW	AES Key Wrap
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
OS	Operating System
PAA	Processor Algorithm Acceleration
PCT	Pairwise Consistency Test
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash 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

FIPS202 SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions
August 2015
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

PKCS#1 Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1
February 2003
http://www.ietf.org/rfc/rfc3447.txt

RFC3394 Advanced Encryption Standard (AES) Key Wrap Algorithm
September 2002
http://www.ietf.org/rfc/rfc3394.txt

RFC5649 Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm
September 2009
http://www.ietf.org/rfc/rfc5649.txt

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

SP800-38E NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices
January 2010
http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf

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

SP800-38G NIST Special Publication 800-38G - Recommendation for Block Cipher Modes of Operation: Methods for Format - Preserving Encryption
March 2016
http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf

SP800-56ARev3 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

SP800-56CRev2 Recommendation for Key Derivation through Extraction-then-Expansion
August 2020
https://doi.org/10.6028/NIST.SP.800-56Cr2

SP800-57 NIST Special Publication 800-57 Part 1 Revision 4 - Recommendation for Key Management Part 1: General
January 2016
http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf

SP800-67 NIST Special Publication 800-67 Revision 1 - Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
January 2012
http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf

SP800-90ARev1 NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators
June 2015
http://dx.doi.org/10.6028/NIST.SP.800-90Ar1

SP800-90B (Second DRAFT) 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

SP800-131A NIST Special Publication 800-131A Revision 1- Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths
November 2015
http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf

SP800-132 NIST Special Publication 800-132 - Recommendation for Password-Based Key Derivation - Part 1: Storage Applications
December 2010
http://csrc.nist.gov/publications/nistpubs/800-132/nist-sp800-132.pdf

SP800-133Rev2 NIST Special Publication 800-133 - Recommendation for Cryptographic Key Generation
June 2020
https://doi.org/10.6028/NIST.SP.800-133r2

SP800-135Rev1 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