Brocade Fabric OS FIPS Cryptographic Module 8.2
FIPS 140-2 Non-Proprietary Security Policy

Document Revision 1.3

May 24, 2019

Prepared for:

Brocade Communications Systems LLC
1320 Ridder Park Drive
San Jose, CA 95131
USA

Prepared By:

www.gossamersec.com

Revision History

Revision	Date	Authors	Summary
1.0	July 20, 2018	Brocade/Gossamer	Release Version
1.1	November 28, 2018	Brocade	A. Added new operational environments to table 8 (the Vendor Affirmed Operational Environments.) B. Corrected misspelled / incorrect kernel reference; 2.16.14.2 was corrected to 2.6.14.2 (reference: table 8) C. Corrected misspelled command reference in section 2.10.1 (`fipscfg`)
1.2	March 12, 2019	Brocade	A. Added new operational environments to table 8 (the Vendor Affirmed Operational Environments.) B. Corrected compiler version for couple of entries for kernel 2.6.14.2 (compiler version was corrected to say “GCC 3.4.6”; reference: table 8)
1.3	May 24, 2019	Brocade	Update to table 8

TABLE OF CONTENTS

1. Introduction

This non-proprietary FIPS 140-2 security policy for the Brocade Fabric OS FIPS Cryptographic Module details the secure operation of the Brocade Communications Systems LLC Brocade Fabric OS FIPS Cryptographic Module as required in Federal Information Processing Standards Publication 140-2 (FIPS 140-2) as published by the National Institute of Standards and Technology (NIST) of the United State Department of Commerce. This document, the Cryptographic Module Security Policy (CMSP), also referred to as the Security Policy, specifies the security rules under which the module must operate.

The Brocade Fabric OS FIPS Cryptographic Module underpins Brocade's Fabric Operating System equipment. The Brocade Fabric OS is the software foundation for Brocade's purpose-built network infrastructure for mission-critical storage. The Brocade Fabric OS family of supported products includes Fiber Channel directors, switches, embedded switches and network extension switches. In addition to supporting the switching functionality of these product lines, Fabric OS supports Fabric Vision Technology features for network monitoring, management, and diagnostics, as well as advanced features that help ensure the highest level of reliability, availability, and serviceability.

2. Brocade Fabric OS FIPS Cryptographic Module 8.2

2.1 Module Specification

The Brocade Fabric OS FIPS Cryptographic Module (hereinafter referred to as the "Library", "cryptographic module" or the "module") is a software only cryptographic module composed of a single shared object (libfipscrypto.so) executing on a general-purpose computer (GPC) system (referred to as "switch hardware" or just "hardware" hereafter) running Brocade's Fabric Operating System.

The physical perimeter of the switch hardware comprises the module's physical cryptographic boundary, while the logical interface of the Brocade Fabric OS FIPS Cryptographic Module shared object constitutes the module's logical cryptographic boundary.

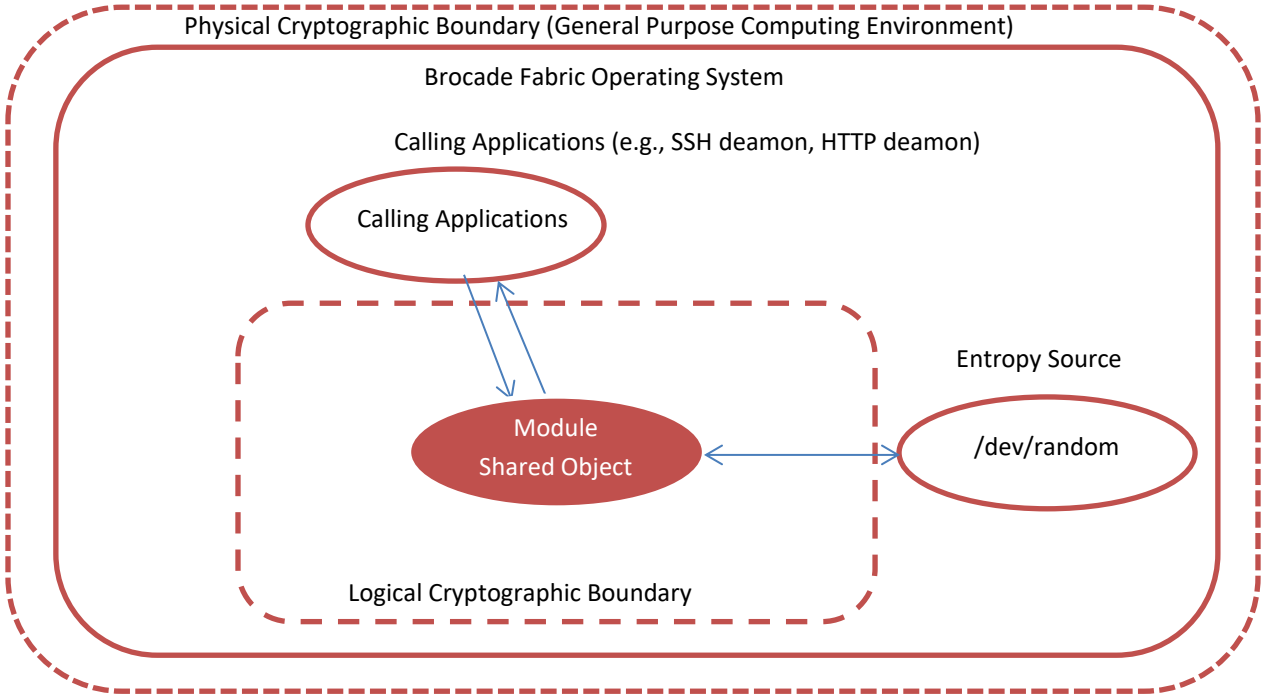

Figure 1 - Logical Diagram

2.1.1 Security Level

The module meets the overall requirements applicable to Level 1 security of FIPS 140-2 and the below specified section security levels.

Table 1 - Module Security Level Specification

#	FIPS 140-2 Section	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self-tests	1
10	Design Assurance	3
11	Mitigation of Other Attacks	N/A
	Overall Level	1

2.1.2 FIPS Mode of Operation

The module provides a comprehensive set of cryptographic algorithms which includes FIPS-Approved algorithms, FIPS-Allowed algorithms, and non-Approved algorithms. A caller wishing to operate the module in a FIPS compliant manner must first configure the module to act as a FIPS module (either in normal mode or in IG 9.11 mode, which is named "9.xx" mode as development occurred while the IG was in draft form), then can call FIPS-Approved and Allowed APIs, and finally must not call any non-Approved APIs. The following tables describe which of the module's services are FIPS-Approved, FIPS-Allowed, and non-Approved.

2.1.3 FIPS-Approved and FIPS-Allowed Cryptographic Algorithms

The module uses cryptographic algorithm implementations that have received the following certificate numbers from the Cryptographic Algorithm Validation Program.

Table 2 – FIPS-Approved Algorithm Certificates

FIPS-Approved Algorithm	CAVP Certificate
AES-128/192/256 ECB/CBC and CFB128	5006
DRBG AES-256 CTR_DRBG	1827
CVL KAS ECC/FFC	1557
ECDSA KeyGen, PKV, Sign/Verify P-256/384/521	1275
HMAC-SHA-1/224/256/384/512	3328
RSA KeyGen, Sig(gen), Sig(ver) 2048/3072	2700
SHA-1/224/256/384/512	4071

Table 3 – FIPS-Allowed Algorithms

FIPS-Allowed Algorithms
Diffie-Hellman (CVL Cert. #1557, key agreement; key establishment methodology provides between 112 and 150 bits of encryption strength)
EC Diffie-Hellman (CVL Cert. #1557, key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength)
NDRNG (used to seed the FIPS-Approved DRBG)
RSA (key wrapping; key establishment methodology provides between 112 and 128 bits of encryption strength)

2.1.4 Non-Approved Cryptographic Algorithms

The module provides the following non-approved cryptographic algorithms. In order to operate the module in a FIPS compliant manner, one cannot call the services (and instead may only utilize FIPS-Approved cryptography). Calling these services would put the module into the non-FIPS mode.

Table 4 – Non-Approved Algorithm s

Non-Approved Algorithm
AES-GCM (non-compliant)
DSA (non-compliant)
TDES (non-compliant)
CAST
Aria
Poly1305
Chacha20
CAMELLIA
SEED
AEAD

2.2 Module Interfaces

The module is classified as a multiple-chip standalone module for FIPS 140-2 purposes. As such, the module's physical cryptographic boundary encompasses the general-purpose computer running Brocade's Fabric Operating System and interfacing with the peripheral devices (USB devices, network devices [Ethernet and Wireless adapters], and power adapter).

However, the module provides only a logical interface via an Application Programming Interface (API) and does not interface or communicate with or across any of the physical ports of the GPC. This logical interface exposes service that calling applications may use directly.

The API interface provided by the module is mapped onto the four FIPS 140-2 logical interfaces: data input, data output, control input, and status output. It is through this logical API that the module

logically separates them into distinct and separate interfaces. The mapping of the module's API to the four FIPS 140-2 interfaces is as follows:

- Data input – input arguments to all functions specifying input parameters
- Data output – modified input arguments (those passed by reference) and return values for all functions modifying input arguments and returning values
- Control input – invocation of all functions
- Status output – information returned by the functions and the output of the RAND_status API (which includes the module's current status)

2.3 Roles, Services and Authentication

The module supports both of the FIPS 140-2 required roles, the Crypto-officer and the User role. An operator implicitly selects the Crypto-officer role when loading (or causing loading of) the library and selects the User role when soliciting services from the module through its API (for example, Fabric OS's SSH daemon acts in the User role when calling the library's API's to obtain cryptographic services). Note that while the Fabric Operating System (Fabric OS) itself provides other roles, these roles are outside the scope of the Fabric OS FIPS Cryptographic Module and thus outside the scope of this security policy. The Fabric OS FIPS Cryptographic Module requires no operator authentication, and the below table enumerates the module's services.

Table 5 - Service Descriptions for Crypto-officer and User Roles

Service	Description
Crypto-Officer services	
Library Loading	The process of loading the shared object/library
User services	
AES_decrypt	AES operations
AES_encrypt	
AES_set_encrypt_key	
DH_OpenSSL	Diffie-Hellman parameter operations
DH_get_default_method	
DH_generate_parameters_ex	
DH_compute_key	
DH_generate_key	
DH_free	
DH_new	
EC_KEY_new	Elliptic Curve operations
EC_KEY_new_by_curve_name	
EC_KEY_generate_key	
EC_KEY_free	
ECDH_compute_key	ECDH Key Agreement
ECDSA_DATA_new_method	ECDSA (signature/verification) operations

Service	Description
ECDSA_do_sign	
ECDSA_do_sign_ex	
ECDSA_do_verify	
ECDSA_sign	
ECDSA_verify	
EVP_Cipher	Envelop (higher-level) function for cipher operations
EVP_Digest	Envelop (higher-level) function for hashing operations
EVP_sha1	
EVP_sha224	
EVP_sha256	
EVP_sha384	
EVP_sha512	
EVP_SignFinal	Envelop (higher-level) function for asymmetric sign/verify
EVP_VerifyFinal	
FIPS_selftest_des_cbc	Self-test functions
FIPS_sha1_test	
FIPS_sha256_test	
FIPS_sha384_test	
HMAC_Final	HMAC operations
HMAC_init	
HMAC_Update	
OPENSSL_clense	Zeroization function used on keying material
OPENSSL_init	Initialization function for the library
RAND_add	DRBG functions
RAND_bytes	
RAND_get_rand_method	
RAND_init_fips	
RAND_load_file	
RAND_poll	
RAND_pseudo_bytes	
RAND_SSLeay	
RAND_status	Module status function
RSA_sign	RSA asymmetric operations
RSA_verify	
RSA_public_encrypt	
RSA_private_decrypt	
RSA_generate_key	
RSA_generate_key_ex	
SHA1_Final	SHA hashing functions
SHA1_Init	
SHA1_Update	
SHA224_Final	
SHA224_Init	
SHA224_Update	

Service	Description
SHA256_Final	
SHA256_Init	
SHA256_Update	
SHA384_Final	
SHA384_Init	
SHA384_Update	
SHA512_Final	
SHA512_Init	
SHA512_Update	

Table 6 - Service Inputs and Outputs

Service	Data Input	Data Output	CSP	Access[1]	Status Out
Crypto-Officer services					
Library Loading	N/A	N/A	N/A	N/A	Flag
User services					
BrocadeCryptoLibraryCryptosystem					
AES_decrypt	ciphertext	plaintext	AES Key	X	Pass/Fail
AES_encrypt	plaintext	ciphertext	AES Key	X	Pass/Fail
AES_set_encrypt_key	AES Key	None	AES Key	W	Pass/Fail
DH_OpenSSL	None	Method (Function list)	N/A	N/A	Pass/Fail
DH_get_default_method	None	Pointer to default method	N/A	N/A	Pass/Fail
DH_generate_parameters_ex	DH context, prime len, generator, callback	DH parameters (PQG)	N/A	G,R	Pass/Fail
DH_compute_key	DH context (private key), peer DH public key	DH shared secret	DH Private key, DH shared secret	X,R	Secret size/0 if fail
DH_generate_key	DH context	DH Private & Public key	DH Private & Public key	G,R	Pass/Fail
DH_free	DH context	None	DH Private & Public key	Z	Pass/Fail
DH_new	None	DH Context	None	N/A	Pass/Fail
EC_KEY_new	None	EC context	N/A	N/A	Pass/Fail

[1] (G)enerate, (R)ead, (W)rite, e(X)ecute, (Z)eroize

Service	Data Input	Data Output	CSP	Access[1]	Status Out
EC_KEY_new_by_curve_name	Curve	EC context	N/A	N/A	Pass/Fail
EC_KEY_generate_key	EC context	EC Pub & Priv key	EC Priv & Pub key	G,R	Pass/Fail
EC_KEY_free	EC context	None	EC Priv & Pub key	Z	Pass/Fail
ECDH_compute_key	EC context, EC Priv key, peer EC Pub key	ECDH shared secret	ECDH secret	G,R	Pass/Fail
ECDSA_DATA_new_method	Engine	Memory allocated for ECDSA data	N/A	N/A	Pass/Fail
ECDSA_do_sign	EC context, Digest	ECDSA signature	EC Priv key	X	Pass/Fail
ECDSA_do_sign_ex	EC context, Digest, EC Priv key	ECDSA signature	EC Priv key	W,X	Pass/Fail
ECDSA_do_verify	EC context Digest, sig, EC Pub key	N/A	EC Pub key	X	Pass/Fail
ECDSA_sign	EC context Digest, EC Priv key	Signature	EC Priv key	X	Pass/Fail
ECDSA_verify	EC context, Digest, sig, EC Pub key	N/A	EC Pub key	X	Pass/Fail
EVP_Cipher	plain text, plain text length	Cipher text	AES key	X	Pass/Fail
EVP_Digest	data to update the context with, data length	output data of EVP_MD_size() length, length of hash	N/A	N/A	Pass/Fail
EVP_sha1	data to update the context with, data length	Message digest structure	N/A	N/A	Pass/Fail
EVP_sha224	data to update the context with, data length	Message digest structure	N/A	N/A	Pass/Fail

Service	Data Input	Data Output	CSP	Access[1]	Status Out
EVP_sha256	data to update the context with, data length	Message digest structure	N/A	N/A	Pass/Fail
EVP_sha384	data to update the context with, data length	Message digest structure	N/A	N/A	Pass/Fail
EVP_sha512	data to update the context with, data length	Message digest structure	N/A	N/A	Pass/Fail
EVP_SignFinal	Message context, RSA or EC Priv key	signature, output length	RSA or EC Priv key	X	Pass/Fail
EVP_VerifyFinal	message context, public key, signature, signature length	N/A	RSA or EC Public key	X	Pass/Fail
FIPS_selftest_des_cbc	N/A	success/failure	N/A	N/A	Pass/Fail
FIPS_sha1_test	N/A	success/failure	N/A	N/A	Pass/Fail
FIPS_sha256_test	N/A	success/failure	N/A	N/A	Pass/Fail
FIPS_sha384_test	N/A	success/failure	N/A	N/A	Pass/Fail
HMAC_Final	hmac context, md	md	HMAC key	X	Pass/Fail
HMAC_init	hash function, key, key length	hmac context	HMAC key	W	Pass/Fail
HMAC_Update	Plain text, plain text length	N/A	HMAC key	X	Pass/Fail
OPENSSL_cleanse	input data pointer, length	output data pointer	All CSP types	Z	Pass/Fail
OPENSSL_init	N/A	N/A	N/A	N/A	Pass/Fail
RAND_add	seed length, seed	Random bytes	DRBG V, DRBG key	W	Pass/Fail
RAND_bytes	Random bytes len	Random bytes	DRBG V, DRBG key	W/X	# of bytes returned

Service	Data Input	Data Output	CSP	Access[1]	Status Out
RAND_get_rand_method	N/A	pointer to random method	N/A	N/A	Pass/Fail
RAND_init_fips	N/A	success/failure	N/A	N/A	Pass/Fail
RAND_load_file	random file, bytes to read	bytes read	DRBG V, DRBG key	W	Pass/Fail
RAND_poll	N/A	success/failure of random number generation using specific entropy source	N/A	N/A	Pass/Fail
RAND_pseudo_bytes	random bytes len	pseudo random bytes	DRBG V, DRBG key	W/X	# of bytes returned
RAND_SSLeay	N/A	returns default RAND_method	N/A	N/A	Pass/Fail
RAND_status	N/A	success/failure status	N/A	N/A	Current module state
RSA_sign	message digest algorithm, message, message len, RSA private key	signature, signature len	RSA private key	W/X	Pass/Fail
RSA_verify	message digest algorithm, message, message len, RSA public key	signature, signature len	RSA public key	X	Pass/Fail
RSA_public_encrypt	Exchanged keying material, RSA public key	ciphertext	RSA public key	X	Pass/Fail
RSA_private_decrypt	Ciphertext, RSA private key	Exchanged keying material	RSA Priv key	X	Pass/Fail
RSA_generate_key	N/A	RSA context with pub/priv key	RSA priv/pub keys	G/W	Pass/Fail

Service	Data Input	Data Output	CSP	Access[1]	Status Out
RSA_generate_key_ex	N/A	RSA context with pub/priv key	RSA priv/pub keys	G/W/R	Pass/Fail
SHA1_Final	message digest, md context	message digest context	N/A	N/A	Pass/Fail
SHA1_Init	message digest context	message digest context	N/A	N/A	Pass/Fail
SHA1_Update	message digest context, input message	message digest context	N/A	N/A	Pass/Fail
SHA224_Final	message digest, md context	message digest context	N/A	N/A	Pass/Fail
SHA224_Init	message digest context	message digest context	N/A	N/A	Pass/Fail
SHA224_Update	message digest context, input message	message digest context	N/A	N/A	Pass/Fail
SHA256_Final	message digest, md context	message digest context	N/A	N/A	Pass/Fail
SHA256_Init	message digest context	message digest context	N/A	N/A	Pass/Fail
SHA256_Update	message digest context, input message	message digest context	N/A	N/A	Pass/Fail
SHA384_Final	message digest context	message digest context	N/A	N/A	Pass/Fail
SHA384_Init	message digest context	message digest context	N/A	N/A	Pass/Fail
SHA384_Update	message digest context	message digest context	N/A	N/A	Pass/Fail
SHA512_Final	message digest, md context	message digest context	N/A	N/A	Pass/Fail
SHA512_Init	message digest context	message digest context	N/A	N/A	Pass/Fail
SHA512_Update	message digest context, input message	message digest context	N/A	N/A	Pass/Fail

2.4 Finite State Model

The module has a Finite State Model (FSM) that describes the module's behavior and transitions based upon its current state and the command received. The module's FSM was reviewed as part of the overall FIPS 140-2 validation.

2.5 Physical Security

The physical security requirements do not apply to the module. The module is a software-only module that executes upon a general-purpose computer.

2.6 Operational Environment

The module executes on a general purpose operating system running in single user mode that segregates processes into separate process spaces. Thus, the operating system separates each process space from all others. The below table lists the specific versions of Fabric Operating System (Fabric OS) upon which validation testing was performed.

Table 7 – Tested Operational Environments

#	Test Platform	Operating System	Kernel	Compiler
1	NXP Semiconductors T1042 (e5500 core) on Brocade G630 Switch	Fabric OS 8.2	2.6.34.6	GCC 4.3.2
2	NXP Semiconductors MPC8548 (e500v2 core) on Brocade DCX 8510-8 Switch	Fabric OS 8.2	2.6.14.2	GCC 3.4.6

In addition, Brocade affirms the module's continued compliance when operating on ABI-compatible operating environments including but not limited to:

Table 8 – Vendor Affirmed Operational Environments

#	Test Platform	Operating System	Kernel	Compiler
1	NXP Semiconductors T1022 (e5500 core) on Brocade G620 Switch	Fabric OS 8.2	2.6.34.6	GCC 4.3.2
2	NXP Semiconductors MPC8548 (e500v2 core) on Brocade DCX 8510-4 Switch	Fabric OS 8.2	2.6.14.2	GCC 3.4.6
3	NXP Semiconductors P3041 (e500mc core) on Brocade 7840 Switch	Fabric OS 8.2	2.6.14.2	GCC 3.4.6
4	NXP Semiconductors P4080 (e500mc core) on Brocade X6-8 Switch	Fabric OS 8.2	2.6.34.6	GCC 4.3.2
5	NXP Semiconductors P4080 (e500mc core) on Brocade X6-4 Switch	Fabric OS 8.2	2.6.34.6	GCC 4.3.2
6	Applied Micro Circuits Corporation (AMCC) 440EPx (e500v2) on Brocade 6510 Switch	Fabric OS 8.2	2.6.14.2	GCC 3.4.6
7	NXP Semiconductors MPC8548 (e500v2 core) on Brocade 6520 Switch	Fabric OS 8.2	2.6.14.2	GCC 3.4.6
8	NXP Semiconductors (Freescale) T1022 (e5500 core) on 16Gb FC Switch Blade for Huawei E9000 (BR 6543)	Fabric OS 8.2.0a	2.6.34.6	GCC 4.3.2

#	Test Platform	Operating System	Kernel	Compiler
9	Applied Micro Circuits Corporation (AMCC) PowerPC 440EPx (e500v2) on FC5022 16Gb SAN Scalable Switch for Lenovo Flex System™ (BR 6547)	Fabric OS 8.2.0a	2.6.14.2	GCC 3.4.6
10	NXP Semiconductors (Freescale) T1022 (e5500 core) on BRCD 16Gb FC Switch for HPE Synergy (BR 6558)	Fabric OS 8.2.0a	2.6.34.6	GCC 4.3.2
11	Applied Micro Circuits Corporation (AMCC) PowerPC 440EPx (e500v2) on Brocade M6505 Fibre Channel Switch for Dell™ PowerEdge™ M1000e (BR M6505)	Fabric OS 8.2.0a	2.6.14.2	GCC 3.4.6
12	NXP Semiconductors T1022 (e5500 core) on Brocade 7810 Extension Switch	Fabric OS 8.2.1	2.6.34.6	GCC 4.3.2
13	NXP Semiconductors (Freescale) T1022 (e5500 core) on Brocade 32Gb Fibre Channel SAN Switch Module for HPE Synergy (BR G648)	Fabric OS 8.2.0_GFT	2.6.34.6	GCC 4.3.2
14	NXP Semiconductors (Freescale) T1022 (e5500 core) on HPE Virtual Connect SE 32Gb FC Module for HPE Synergy	Fabric OS 8.2.0_CBN	2.6.34.6	GCC 4.3.2

Please note that the CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment which is not listed on the validation certificate.

2.7 Key Management

The module possesses only one key, its self-integrity test HMAC key. Beyond that key, the module does not store any other keys persistently, and it is the calling applications responsibility to appropriately manage keys. The module cannot generate keys but can accept keys entered by an operator, and affords an operator the ability to zeroize keys held in RAM. The following table describes the module's Critical Security Parameters (CSPs) including asymmetric and symmetric keys.

Table 9 - Module Keys/CSPs

Key	Type	Size	Description	Origin	Stored	Zeroized
AES key	AES	128/ 192/ 256	Symmetric keys used for encryption & decryption	Entered by calling application	RAM / plaintext	Zeroize context
DH Parameters	DH	2048-4096	PQG tuplet	Entered by calling application or generated by module	RAM / plaintext	Zeroize context
DH Private & Public Key	DH	112-256, 2048-4096	Asymmetric keys used for key exchange	Entered by calling application or generated by module	RAM / plaintext	Zeroize context
DH Shared-Secret	Secret	2048-4096	Shared Secret resulting from the DH exchange	Key agreement	RAM / plaintext	Zeroize context
DRBG key	AES-256	256-bits	Internal state of the CTR_DRBG	Seeding from DRBG entropy	RAM / plaintext	Module unload
DRBG V	DRBG	128-bits	Internal state of the CTR_DRBG	Seeding from DRBG entropy	RAM / plaintext	Module unload

Key	Type	Size	Description	Origin	Stored	Zeroized
DRBG entropy	random	384-bits	Used to instantiate, reseed, or add to CTR_DRBG	/dev/random	RAM / plaintext	After use
EC Private & Public Key	EC	Curves P-256, 384, 521	Asymmetric keys used for key exchange or for signatures	Entered by calling application or generated by module	RAM / plaintext	Zeroize context
ECDH Shared-Secret	Secret	256, 384, 521	Shared Secret resulting from the ECDH exchange	Key agreement	RAM / plaintext	Zeroize context
HMAC Key	HMAC	112-512 bits	Secret key used for HMAC-SHA computation	Entered by calling application	RAM / plaintext	Zeroize context
RSA Private & Public Key	RSA	2048-4096	Asymmetric keys used for signature generation or key exchange	Entered by calling application or generated by module	RAM / plaintext	Zeroize context
Self-integrity HMAC Key	HMAC	128-bits	HMAC key used by the module for its power up integrity test	Compiled into the module	Module image	N/A (see 140-2 IG 7.4)

2.8 Electromagnetic Interference and Compatibility

The module meets level 1 security for FIPS 140-2 EMI/EMC requirements as the Brocade Fabric OS FIPS Cryptographic Module passed validation executing upon general-purpose computers that conform to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart A, Unintentional Radiators, Digital Devices, Class A (i.e., for office use).

2.9 Self-Tests

Upon installation and configuration by the Crypto-officer, the module automatically performs either a complete set of power-up self-tests during library load to ensure proper operation or performs an initial set of complete power-up self-tests after boot and (assuming those tests pass) thereafter performs only its integrity test in accordance with IG 9.11.

1. Power-On Self-Tests:
    a. AES encryption and decryption KATs
    b. ECDSA Pairwise Consistency Test (PWCT)
    c. RSA (sign/verify) KATs
    d. SP 800-90A CTR_DRBG KAT (Note: DRBG Health Tests as specified in SP800-90A Section 11.3 are performed)
    e. HMAC KATs (HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512)
    f. SHA KATs (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)
    g. ECC DH Primitive Z Computation KAT
    h. Software Integrity Check (HMAC-SHA256)

2. Conditional Tests

a. Continuous Random Number Test of the SP800-90A DRBG
b. Continuous Random Number Test of the NDRNG
c. RSA Pairwise Consistency Test
d. ECDSA Pairwise Consistency Test

An operator has no access to cryptographic functionality unless the cryptographic module self-tests pass and the library load succeeds. The power-up self-tests include an integrity check of the module's software using verification of an HMAC signature calculated over the module's file image. Should the module fail a self-test, the module will return an error and inhibit all cryptographic operations. Finally, an operator may invoke all power-up self-tests at any time by power-cycling the GPC and then reloading the module.

2.10 Guidance and Secure Operation

The Module meets overall Level 1 requirements for FIPS PUB 140-2. In accordance with 140-2 Implementation Guidance 7.14 section 1.(b) the module gets a minimum of 384-bits of entropy (contained within 384-bits of data) from module call to the NDRNG.

The sections below describe the Crypto-officer and User guidance.

2.10.1 Crypto-officer Guidance

The Crypto-officer or operator responsible for configuring the operational environment upon which the module runs must ensure FIPS compliant operation (as described in section 2.1.2, FIPS Mode of Operation, of the Security Policy).

The Crypto-officer can configure the module to operate in a FIPS compatible mode by modifying the module's configuration flag in the overall Fabric OS configuration by executing the fipscfg CLI command available with Fabric OS as shown below.

To Show FIPS Mode

```
fipscfg --show
```

To Enable FIPS Inside Mode

```
fipscfg --enable fipsinside
```

To Enable FIPS Inside IG 9.11 draft mode

```
fipscfg --enable fipsinside -9.xx
```

To Disable FIPS Inside Mode

```
fipscfg –disable fipsinside
```

To Disable FIPS Inside IG 9.11 draft mode

```
fipscfg --disable fipsinside -9.xx
```

or

```
fipscfg --disable fipsinside
```

Additionally, the Crypto-officer is defined to be the operator responsible for loading the library, thus when invoked by a calling application (either at library load or dynamically), the operating system loader

will load the module, causing it to automatically perform its power-up self-tests. Should the module fail its power-up self-tests, the module sets a status indicator and inhibits its cryptographic functions.

2.10.2 User Guidance

Once the operating system has been properly configured by the Crypto-officer (if needed), a user (calling application) of the Brocade Fabric OS FIPS Cryptographic Module must adhere to the rules of section 2.1.2 and only call FIPS-Approved and Allowed services/APIs in order to operate the module in a FIPS-compliant manner. The module utilizes only FIPS-Approved cryptographic algorithms. The calling application must assume responsibility for managing keys, as the module does not provide any persistent key storage.

2.11 Mitigation of Other Attacks

The Brocade Fabric OS FIPS Cryptographic Module does not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for validation.