Fungible, Inc.

FunOS Crypto Module

FIPS 140-2 Non-Proprietary Security Policy

Software Version: 1.0.0

Hardware Versions: F1 rev A0 and S1 rev A0

Date: December 1, 2022

Prepared for:

Fungible, Inc.

3201 Scott Blvd.

Santa Clara, CA 95054
United States of America

www.fungible.com

Prepared by:

Acumen Security, LLC.

2400 Research Blvd.
Suite 395
Rockville, MD 20850
United States of America

Phone: +1 703 375 9820
www.acumensecurity.net

Introduction

Federal Information Processing Standards Publication 140-2 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) run the FIPS 140 program. The NVLAP accredits independent testing labs to perform FIPS 140 testing; the CMVP validates modules meeting FIPS 140 validation. Validated is the term given to a module that is documented and tested against the FIPS 140 criteria.

More information is available on the CMVP website at: https://csrc.nist.gov/projects/cryptographic-module-validation-program

About this Document

This non-proprietary Cryptographic Module Security Policy for the FunOS Crypto Module from Fungible, Inc. provides an overview of the product and a high-level description of how it meets the overall Level 1 security requirements of FIPS 140-2.

The FunOS Crypto Module may also be referred to as the “module” in this document.

Notices

Table of Contents

Introduction 2

About this Document 2

Notices 2

1. Introduction 5

2. FIPS 140-2 Security Levels 7

3. Cryptographic Module Specification 8

3.1 Cryptographic Boundary 8

4. Modes of Operation 9

5. Cryptographic Module Ports and Interfaces 10

6. Roles, Services and Authentication 10

7. Physical Security 12

8. Operational Environment 12

9. Cryptographic Algorithms & Key Management 13

9.1 Approved Cryptographic Algorithms 13

9.2 Allowed Cryptographic Algorithms 14

9.3 Non-Approved Cryptographic Algorithms 14

9.4 Cryptographic Key Management 14

9.5 Key Generation and Entropy 15

9.6 Key Storage 15

9.7 Key Zeroization 16

10. Self-tests 16

10.1 Power-On Self-Tests 16

10.2 Conditional Self-Tests 17

11. Mitigation of Other Attacks 17

12. Security Rules and Guidance 18

12.1 Usage of AES-GCM 18

12.2 AES-XTS Key validation 18

13. References and Standards 19

14. Acronyms and Definitions 20

List of Tables

Table 1 – Tested Operational Environments 5
Table 2 – Vendor Affirmed Operational Environments 5
Table 3 – Validation Level by FIPS 140-2 Section 7
Table 4 – Components 8
Table 5 – Ports and Interfaces 10
Table 6 – Approved Services, Roles and Access Rights 11
Table 7 – non-Approved or non-security relevant services 11
Table 8 – Approved Algorithms and CAVP Certificates (AES Engine) 13
Table 9 – Approved Algorithms and CAVP Certificates (SHA Engine) 14
Table 10 – Approved Algorithms and CAVP Certificates (TRNG) 14
Table 11 – Allowed Algorithms 14
Table 12 – non-Approved Algorithms 14
Table 13 – Keys and CSPs 15
Table 14 – Power-on Self-Tests 17
Table 15 – Conditional Self-Tests 17
Table 16 – References and Standards 19
Table 17 – Acronyms 20

List of Figures

Figure 1 – S1 DPU Chip 6
Figure 2 – F1 DPU Chip 6
Figure 3 – FunOS Crypto Module Block Diagram 9

1. Introduction

Fungible, Inc. FunOS Crypto Module (hereafter referred to as the "module") runs as part of the FunOS operating system. This module has two parts – libfuncrypto.a, a software library providing APIs for AES, Digest, HMAC, and DRBG operations, and a disjoint hardware chip-component (SEC-F1 or SEC-S1 contained within the Fungible DPU) containing the cryptographic algorithm implementations called from the software APIs. The SEC component can be invoked only from the software APIs and there are no other interfaces available to access them. The validated version of the module is 1.0.0. The module is a Software-Hybrid Module type with a Multi-Chip Standalone Embodiment, per the FIPS 140-2 standard.

The cryptographic module was tested on the following operating environments detailed below:

#	Operating System	Processor (DPU)	Platform
1	FunOS 3.2.1	Fungible F1 rev A0 with MIPS64 (chip contains SEC-F1) with PAI	FS-1600
2	FunOS 3.2.1	Fungible S1 rev A0 with MIPS64 (chip contains SEC-S1) with PAI	FC-200

Table 1 – Tested Operational Environments

The cryptographic module is also supported on the following operating environments for which operational testing and algorithm testing was not performed:

#	Operating System	Processor (DPU)	Product Family
1	FunOS 3.2.1	F1 rev A0	FS-800
2	FunOS 3.2.1	S1 rev A0	FC-50
3	FunOS 3.2.1	S1 rev A0	FC-100

Table 2 – Vendor Affirmed Operational Environments

As per FIPS 140-2 Implementation Guidance G.5, compliance is maintained for other versions of the respective operational environments where the module is unchanged. No claim can be made as to the correct operation of the module or the security strengths of the generated keys if the module is modified.

FIPS 140-2 validation compliance is maintained when the module is operated on other versions of the chip running in single-user mode, assuming that the requirements outlined in FIPS 140-2 IG G.5 are met.

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.

The following figure shows the S1 DPU chip:

Figure 1 – S1 DPU Chip

Figure 2 below shows the F1 DPU chip:

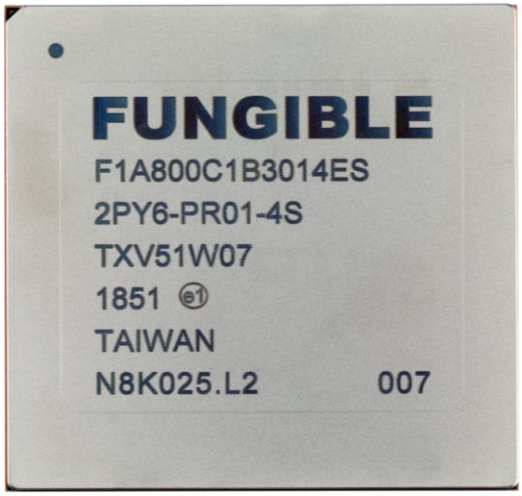

Figure 2 – F1 DPU Chip

2. FIPS 140-2 Security Levels

The following table lists the level of validation for each area in FIPS 140-2:

FIPS 140-2 Section Title	Validation Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	1
Operational Environment	1
Cryptographic Key Management	1
Electromagnetic Interference / Electromagnetic Compatibility	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A
Overall Level	1

Table 3 – Validation Level by FIPS 140-2 Section

3. Cryptographic Module Specification

3.1 Cryptographic Boundary

The module is comprised of a software library and a disjoint hardware accelerator component with cryptographic engines. The accelerator component is contained within the Fungible DPU (F1 or S1). The accelerator can be invoked by the APIs to perform cryptographic operations. All operations of the module occur via API calls from the FunOS operating system to the FunOS Cryptographic Module.

The cryptographic logical boundary of the module consists of the cryptographic software library and the hardware accelerator in which the software library interfaces with.

Component	Description
libfuncrypto.a	This library provides APIs for cryptographic algorithms. These APIs internally schedule cryptographic operations as a “work unit” on a hardware accelerator.
Fungible S1 containing the SEC-S1 Hardware Accelerator Or Fungible F1 containing the SEC-F1 Hardware Accelerator	Multi-threaded crypto accelerator which includes SHA, AES, HMAC and DRBG engines

Table 4 – Components

The module performs no communications other than with the calling application, i.e., through the FunOS operating system APIs as shown in Figure 3 below.

The software module executes entirely within the DPU (F1 or S1) . The module executes on the F1/S1 integrated MIPS DPU cores. The F1 has 52 MIPS cores, and the S1 has 16 MIPS cores. The F1/S1 both have internal memory (SRAM) and externally attached memory (HBM (F1 only) and DDR4).

Figure 3 (below) shows the logical relationship of the cryptographic module to the other software and hardware components of the host platform:

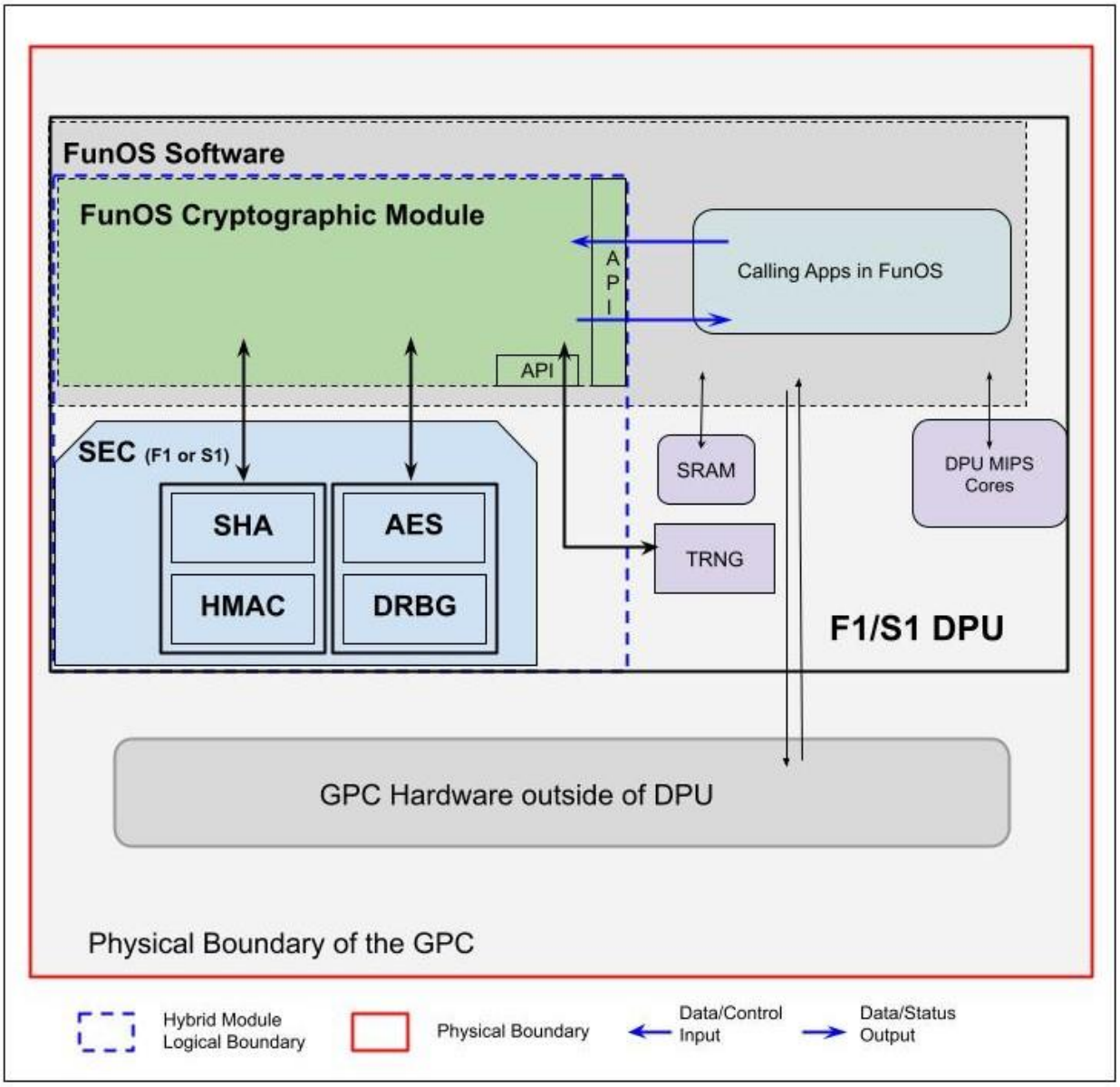

Figure 3 – FunOS Crypto Module Block Diagram

4. Modes of Operation

The module supports two modes of operation: FIPS Approved and non-Approved. The module will be in FIPS Approved mode when all power-on Self-Tests have completed successfully, and only Approved or allowed algorithms are invoked. See Tables 8 through 11 (below) for a list of the supported Approved and Allowed algorithms. The non-Approved mode is entered when a non-Approved algorithm is invoked. See Table 12 for a list of non-Approved algorithms.

5. Cryptographic Module Ports and Interfaces

The Data Input interface consists of the input parameters of the API functions. The Data Output interface consists of the output parameters of the API functions. The Control Input interface consists of the actual API input parameters. The Status Output interface includes the return values of the API functions.

FIPS Interface	Logical Interfaces
Data Input	Input parameters of API calls
Data Output	Output parameters of API calls
Control Input	API calls
Status Output	Return values of API calls
Power Input	Physical power connector on the DPU is powered from the circuit board.

Table 5 – Ports and Interfaces

All status ports and control ports are directed through the interface of the FunOS Crypto Module's logical boundary (libfuncrypto), which is its software APIs. The Crypto Officer interacts with the library in two distinct ways:

- Initializing the module; and
- The application services (APIs) invoked by users.

The User interacts with the library through the application services (APIs) invoked.

Data input and data output are provided in the variables passed in the API and callable service invocations, generally through caller-supplied buffers. Control inputs are provided through dedicated parameters. Status output is provided in return codes and through messages.

All the data output is inhibited during self-test, FIPS error or zeroization. As a software-hybrid module, control of the physical ports is outside the module scope. However, when the module is performing self-tests, or is in an error state, all output on the module's logical data output interfaces is inhibited.

6. Roles, Services and Authentication

The cryptographic module implements both User and Crypto Officer (CO) roles. The module does not support user authentication. The User and CO roles are implicitly assumed by the entity accessing services implemented by the module. A user is considered the owner of the thread that instantiates the module and, therefore, only one concurrent user is allowed.

The Approved services supported by the module and access rights within services accessible over the module's public interface are listed in the table below:

- Generate: Generates the Critical Security Parameter using an approved Random Bit Generator

- Read: Export the CSP

- Write: Enter/establish and store a CSP

- Zeroize: Overwrite the CSP

- Execute: Employ the CSP

Service	Description	Roles	Keys and/or CSPs	Access rights to keys/CSPs
Module Initialization	Validate signatures of images, POST	CO	N/A	N/A
Symmetric encryption/decryption	Encrypts or decrypts a block of data using AES	User, CO	AES Key AES-GCM Key AES-XTS key	Write/ Execute
Keyed hashing	Keyed-Message Digest Operations	User, CO	HMAC key	Write/ Execute
Symmetric Key Generation	Generating the AES and HMAC Keys	User, CO	AES Key AES-GCM Key AES-XTS key HMAC key	Generate
Hashing (SHS)	Message Digest Operations	User, CO	N/A	N/A
Self-test	Performing the self-tests by power-cycling (reboot)	User, CO	N/A	N/A
Zeroization	Zeroizes CSPs	User, CO	All keys	Zeroize
Show status	Displays status of the module	User, CO	N/A	N/A
DRBG	Generate a string of pseudo-random numbers	User, CO	Entropy input string, DRBG internal state	Execute
TLS Key Derivation	Derives a TLS master secret from the TLS Pre-master secret	User, CO	TLS Pre-master secret, TLS master secret	Read/Write /Execute

Table 6 – Approved Services, Roles and Access Rights

The module provides the following non-Approved services, which utilize algorithms listed in Table 12:

Service	Non-Approved Functions
Chacha, Poly (Available on the S1 DPU only)	Chacha20, Poly1305, Chacha20+Poly1305 encryption and hashing functions

Table 7 – non-Approved or non-security relevant services

7. Physical Security

The FunOS Crypto Library is a software-hybrid module implemented as part of Fungible's F1 and S1 Data Processing Units (DPUs).. The hardware components of the module, Fungible's F1 and S1 Data Processing Units (DPUs), have a production-grade enclosure and hence conform to the Level 1 requirements for physical security.

8. Operational Environment

The module operates in a modifiable operational environment as per FIPS 140-2 Security Level 1 specifications. The module runs on the FunOS operating system executing on the hardware specified in Table 1.

9. Cryptographic Algorithms & Key Management

9.1 Approved Cryptographic Algorithms

The module implements the following FIPS 140-2 Approved algorithms:

CAVP Cert #	Algorithm	Standard	Mode/Method/Size	Use
A2325 A2327	AES	FIPS 197 SP 800-38A	128, 192, 256 CBC, ECB, CTR	Encryption, Decryption
A2325 A2327	AES	SP 800-38C	128, 192, 256 CCM	Encryption, Decryption
A2325 A2327	AES	SP 800-38D	128, 192, 256 GCM	Authenticated Encryption, Authenticated Decryption
A2325 A2327	AES[1]	SP 800-38E	128, 256 XTS	Encryption, Decryption
Vendor Affirmed	CKG	SP800-133r2 Section 4	256-bit	Cryptographic Key Generation
A2325 A2327	DRBG	SP 800-90Ar1	AES-256 CTR_DRBG	Random Bit Generation

Table 8 – Approved Algorithms and CAVP Certificates (AES Engine)

CAVP Cert #	Algorithm	Standard	Mode/Method/Size	Use
A2326 A2328	SHA	FIPS 180-4	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	Digital Signature Generation, Digital Signature Verification, non-Digital Signature Applications
A2326 A2328	SHA3	FIPS 202	SHA3-224, SHA3-256, SHA3-384, SHA3-512	Digital Signature Generation, Digital Signature Verification, non-Digital Signature Applications
A2326 A2328	HMAC	FIPS 198-1	HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/224, HMAC-SHA2-512/256, HMAC-SHA3-224,	Generation, Authentication

[1] AES-XTS shall only be used in storage applications.

CAVP Cert #	Algorithm	Standard	Mode/Method/Size	Use
			HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512	
A2326 A2328	SHAKE	SP 800-185	SHAKE-128, SHAKE-256	
A2326 A2328	(CVL) KDF TLS 1.2[2]	SP 800-135r1	PRF	Key Derivation

Table 9 – Approved Algorithms and CAVP Certificates (SHA Engine)

CAVP Cert #	Algorithm	Standard	Mode/Method/Size	Use
N/A	ENT (P)	SP 800-90B	Conditioning Component - AES-CBC-MAC	Entropy Source

Table 10 – Approved Algorithms and CAVP Certificates (TRNG)

9.2 Allowed Cryptographic Algorithms

The module employs the methods listed in Table 11, which are allowed for use in a FIPS-Approved mode but no security claimed.

Algorithm	Use
GHASH[3] (within GCM)	Message Authentication. No security claimed.

Table 11 – Allowed Algorithms

9.3 Non-Approved Cryptographic Algorithms

The module employs the methods listed in Table 12, which are not allowed for use in a FIPS-Approved mode. Their use will result in the module operating in a non-Approved mode.

Algorithm	Use
Chacha20, Poly1305, Chacha20+Poly1305	Chacha20 – Used for Encryption/Decryption Poly1305 – Used for hashing

Table 12 – non-Approved Algorithms

9.4 Cryptographic Key Management

The table below provides a complete list of Private Keys and CSPs used by the module:

Key/CSP Name	Key Description	Generated/ Input	Output
AES Key	AES CBC, ECB, CTR encrypt /	Generated internally by the DRBG or	N/A

[2] The module does not implement the Key Agreement schemes associated with the TLS KDF. The protocols associated with the KDF have not been reviewed or tested by CAVP or CMVP.

[3] GHASH is permitted for use in the FIPS-Approved mode with no security claimed per IG 1.23 scenario 3.

Key/CSP Name	Key Description	Generated/ Input	Output
	decrypt key (128/192/256)	Input via API in plaintext	
AES-GCM Key	AES encrypt / decrypt (128/192/256)	Generated internally by the DRBG or Input via API in plaintext	N/A
AES-XTS Key	AES encrypt / decrypt key (128/256)	Generated internally by the DRBG or Input via API in plaintext	N/A
HMAC Key	Keyed hash key (160/224/256/384/512)	Generated internally by the DRBG or Input via API in plaintext	N/A
Entropy input string	Entropy input strings used as seed to the DRBG. 384 bits	From the output of the TRNG	N/A
DRBG Internal state (V, Key)	Used to generate random bits. V = 128 bits. Key = 256 bits.	During DRBG initialization.	N/A
TLS Pre-master Secret	Secret value used as input in the TLS PRF. 48 bytes of pseudorandom data	Input via API in plaintext	N/A
TLS Master Secret	Shared secret. 48 bytes of pseudorandom data	Derived from pre-master secret using the SP 800-135 TLS KDF.	Output via API in plaintext

Table 13 – Keys and CSPs

9.5 Key Generation and Entropy

The module is a software-hybrid module which includes one hardware-based CTR_DRBG conformant to SP 800-90A, which is seeded by an SP 800-90B compliant entropy source. The module's DRBG is seeded with 384 bits.

The module's entropy source is consistent with Scenario 1 (b) described in FIPS 140-2 IG 7.14. The module generates symmetric keys in accordance with FIPS 140-2 IG D.12 and SP 800-133rev2, Section 4.

The module performs the health tests for the SP 800-90A DRBG as defined per Section 11.3 of SP 800-90A.

9.6 Key Storage

The cryptographic module does not perform persistent storage of keys. Keys and CSPs are passed to the module by the calling application with the exception of the entropy string, which is passed to the DRBG by libfuncrypto after obtaining from the TRNG source in the physical boundary. The keys and CSPs are stored in volatile memory in plaintext. Keys and CSPs residing in internally allocated data structures

(during the lifetime of an API call) can only be accessed using the module-defined API. The operating system protects memory and process space from unauthorized access.

9.7 Key Zeroization

The module is passed keys as part of a function call from a calling application and does not store keys persistently. The calling application is responsible for parameters passed in and out of the module. The Operating System and the calling application are responsible to clean up temporary or ephemeral keys.

All CSPs can be zeroized by power-cycling or rebooting the host platform (DPU chip).

10. Self-tests

FIPS 140-2 requires the module to perform self-tests to ensure the integrity of the module and the correctness of the cryptographic functionality at start-up. The supported tests are listed and described in this section.

10.1 Power-On Self-Tests

Power-on self-tests are run upon the initialization of the module and do not require operator intervention to run. If any of the tests fail, the module will not initialize. The module will enter an error state and no services can be accessed.

The module implements the following power-on self-tests:

Type	Test
Integrity Test	HMAC-SHA-256
Known Answer Test (AES Engine)	AES ECB 128, 192, 256 Encrypt and Decrypt KATs
	AES GCM 128, 192, 256 Encrypt and Decrypt KATs
	AES XTS 128, 256 Encrypt and Decrypt KATs
	DRBG Instantiate, Reseed, Generate KAT (per SP 800-90A Section 11)
Known Answer Test (SHA Engine)	HMAC-SHA-1 KAT
	HMAC-SHA-256 KAT
	HMAC-SHA-512 KAT
	HMAC-SHA3-256 KAT
	SHAKE-SHA3-256 KAT
	TLS KDF KAT

Table 14 – Power-on Self-Tests

By default, all power-on self-tests are executed at module initialization. The power-on self-tests can be run on demand by power-cycling the host platform.

For the integrity test, at build time (at the factory), an HMAC value is computed over the full contents of the ".text" and ".rodata" sections of the completed FunOS Crypto Module image as determined by the ELF headers and is appended to the image in a distinct segment "fipsmac". At boot time code in the FunOS Crypto Module retrieves the boundaries of the ".text" and ".rodata" sections by using variables provided by the linker. It then computes a HMAC value using its internal routines over the full content of the ".text" and ".rodata" sections and compares it to the value placed in the "fipsmac" segment. If the values differ, the chip is reset, reboots and the FunOS image is loaded again (and executes the integrity test as part of its initialization).

10.2 Conditional Self-Tests

Conditional self-tests are run under specific conditions, such as during instantiation or when a random value is requested from the DRBG.

Type	Test
CRNGT for DRBG	DRBG Continuous Test performed when a random value is requested from the DRBG.
DRBG Health Checks	Performed on DRBG, per SP 800-90A Section 11.3. Required per IG C.1
SP 800-90B Health Tests	Adaptive Proportion Test and Repetition Count Test required per NIST SP 800-90B

Table 15 – Conditional Self-Tests

In the event that any of the above conditional tests fail the module will transition to an error state where no services can be accessed by operators.

11. Mitigation of Other Attacks

The module is not designed to mitigate against attacks that are outside of the scope of FIPS 140-2.

12. Security Rules and Guidance

The module design corresponds to the Module security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-2 Level 1 module.

1. The module does not provide authentication.
2. The operator shall be capable of commanding the module to perform the power-on self-tests by cycling power.
3. Power-on self-tests do not require any operator action.
4. Data output shall be inhibited during self-tests, zeroization, and error states.
5. Output related to keys and their use is inhibited until the key concerned has been fully generated.
6. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
7. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
8. The module does not support concurrent operators.
9. The module does not have any external input/output devices used for entry/output of data.
10. The module does not enter or output plaintext CSPs from the module's physical boundary.
11. The module does not output intermediate key values.

12.1 Usage of AES-GCM

The AES-GCM IV for encryption is constructed deterministically per Section 8.2.1 of NIST SP 800-38D. The AES-GCM IV construction is performed within the module boundary in accordance with FIPS 140-2 IG A.5 scenario 3 using a deterministic method. The IV is constructed using 64-bits from the salt and name (fixed fields) allowing for 2^{32} possible values and a 32-bit incremental counter value to form a 96 bit IV.

The implementation of the deterministic non-repetitive counter management logic inside the module ensures that when the counter portion of the IV exhausts the maximum number of possible values for a given session key, the encryptor aborts the session.

If the module power is lost and restored, the calling application must ensure that any AES-GCM keys used for encryption or decryption are redistributed.

12.2 AES-XTS Key validation

To meet the requirement stated in FIPS 140-2 IG A.9, the module implements a check to ensure that the two AES keys used in AES XTS mode are not identical. The module checks explicitly that Key_1 ≠ Key_2, and the check for Key_1 ≠ Key_2 is done before using the keys in the XTS-AES algorithm to process data with them.

13. References and Standards

The following Standards are referred to in this Security Policy:

Abbreviation	Full Specification Name
FIPS 140-2	Security Requirements for Cryptographic Modules
FIPS 180-4	Secure Hash Standard (SHS)
FIPS 197	Advanced Encryption Standard
FIPS 198-1	The Keyed-Hash Message Authentication Code (HMAC)
FIPS 202	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions
IG	Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program
SP 800-38A	Recommendation for Block Cipher Modes of Operation
SP 800-38C	Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality
SP 800-38D	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC
SP 800-38E	Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices
SP 800-90Ar1	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
SP 800-90B	Recommendation for the Entropy Sources Used for Random Bit Generation
SP 800-133r2	Recommendation for Cryptographic Key Generation
SP 800-135r1	Recommendation for Existing Application-Specific Key Derivation Functions
SP 800-185	SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash, and ParallelHash

Table 16 – References and Standards

14. Acronyms and Definitions

The following table lists the acronyms and abbreviations used in this document.

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher-Block Chaining
CCCS	Canadian Centre for Cyber Security
CMVP	Crypto Module Validation Program
CO	Cryptographic Officer
CSP	Critical Security Parameter
CTR	Counter-mode
DPU	Data Processing Unit
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standards
GCM	Galois/Counter Mode
GMAC	Galois Message Authentication Code
HMAC	Key-Hashed Message Authentication Code
IG	Implementation Guidance
KAT	Known Answer Test
LLC	Limited Liability Company
N/A	Not Applicable
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Lab Accreditation Program
RAM	Random Access Memory
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SP	Special Publication

Table 17 – Acronyms