Fatpipe Crypto Module

FIPS 140-2 Non-Proprietary
Security Policy

Revision History

Date	Version	Changes
07/25/2017	1.0	Initial Draft

Table of Contents

1 Module Overview

The following summarize key features of the Fatpipe Crypto Module (Software Version: 9.1.2-fips). Herein, the Fatpipe Crypto Module may be referred to as “the module”.

The module is a software-only cryptographic module designed to provide router clustering. It is an essential part of Disaster Recovery and Business Continuity Planning for Virtual Private Network (VPN) connectivity. It is integrated with several User Space and Kernel Space cryptographic algorithms and other security mechanisms.

The module is a multichip standalone module.

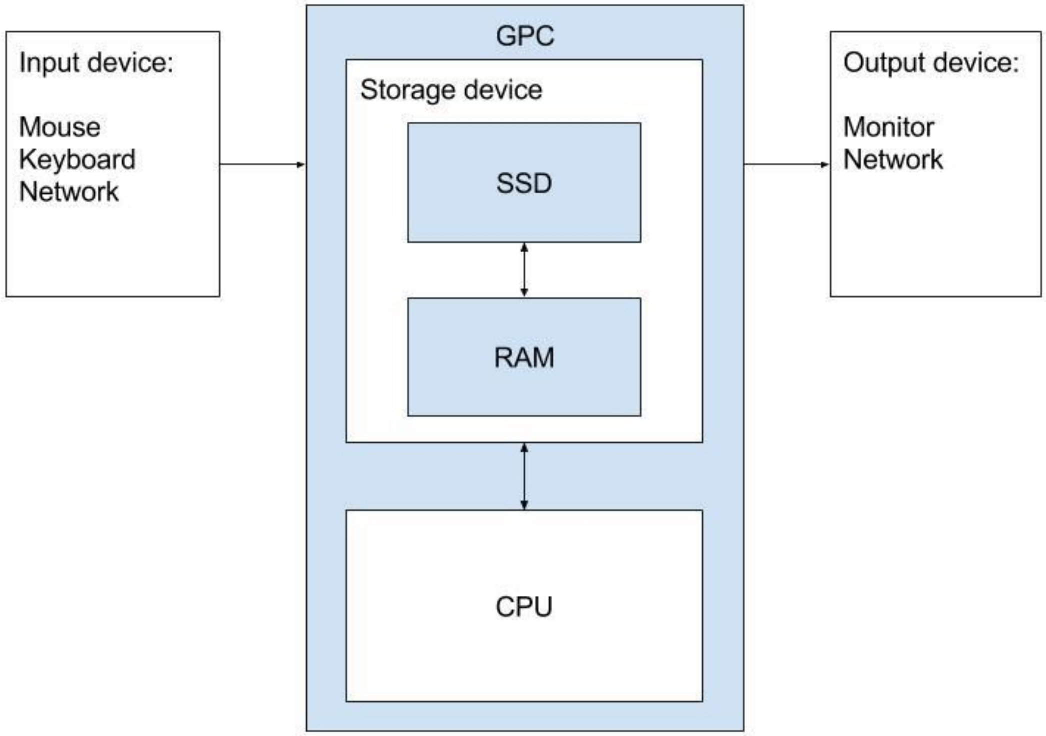
The module is tested under the configuration below:

Table 1: Tested Configurations

Operating System	Processor	Optimizations
LFS (Linux from scratch) 1.1.0 x86 64 Pure 64	Intel(R) Xeon(R) CPU E3-1220 v5 @ 3.00GHz without AES-NI	None

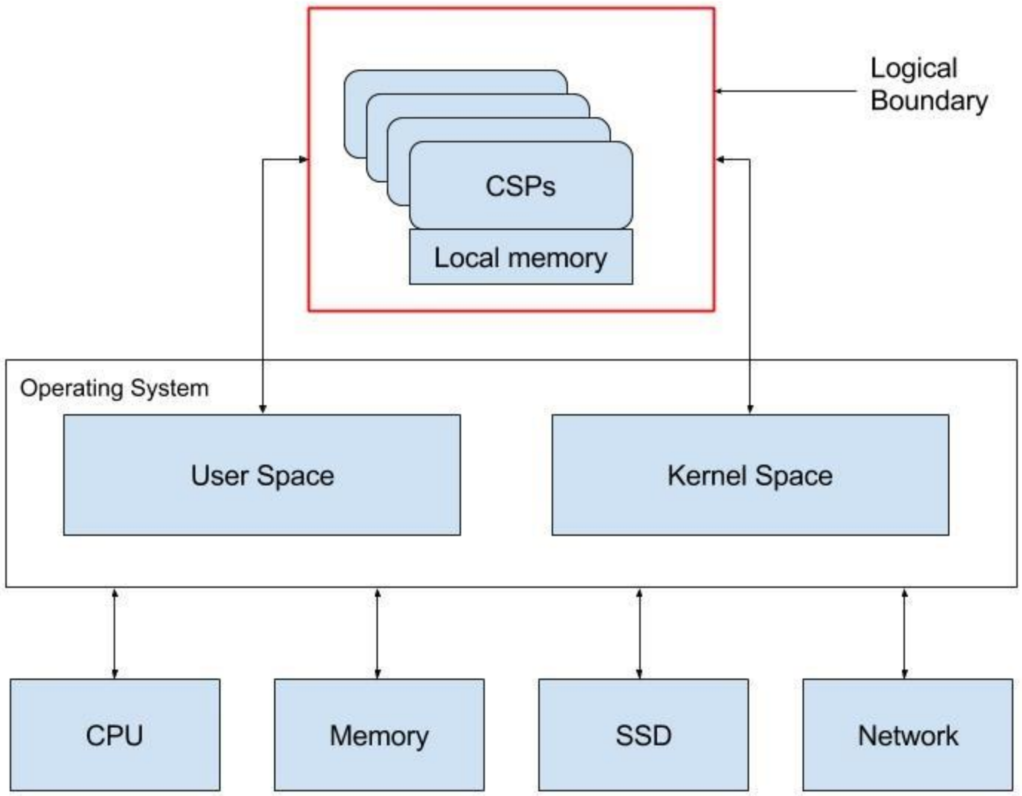
1.1 Cryptographic Boundary

1.1.1 Hardware Block Diagram



1.1.2 Software Block Diagram

The logical boundary of the module is shown below.



…

2 Security Level

The module meets the overall requirements applicable to Security Level 1 of FIPS 140-2, as shown below.

Table 2: Security Level Summary

Security Requirements	**Level**
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

3 Ports and Interfaces

The physical ports of the general-purpose computer on which the module runs, such as keyboards, hard-disks, displays, etc., provide a means to access the module. The physical ports of the general purpose computer that were tested are listed here:

- 4-port Gigabit NIC
- (Qty. 2) USB ports
- Serial Console port
- Power Port
- VGA Port

The module does not support a Maintenance interface. The logical interfaces are described below.

Table 3: Module Logical Interfaces

Interface	Description
Data Input	Via API
Control Input	Via API
Data Output	Via API
Status Output	Via API

4 Security Rules

The following specifies the security rules under which the module shall operate:

- Installation of the module is the responsibility of the Administrator.
- The module enforces logical separation between all data inputs, data outputs, control inputs, and status outputs.
- All data output is inhibited when in an error state, during self-tests, and during zeroization.
- Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
- The general purpose-computing platform includes a power port.
- The module protects public keys and CSPs from unauthorized disclosure, unauthorized modification, and unauthorized substitution.
- Power-up self-tests do not require any operator intervention.
- The module does not support bypass capability.
- The module does not output intermediate key values.
- The module does not support a maintenance interface or role.
- When performing zeroization, the operator of the module is **required** to reformat and overwrite the drive completely, and the operator **must** reboot the module.

5 Modes of Operation

The module supports two modes of operation: FIPS Approved mode and non-Approved mode.

5.1 FIPS Approved Mode

The Crypto Officer shall follow these steps to initialize the module to run in the FIPS Approved mode:

1. Power on the module
2. Load Linux kernel module
    a. In init function, fips_test is called to perform self-tests without operator intervention.
3. After self-tests are passed, the module is in FIPS Approved mode, user will see "ALL TESTS [PASSED]. ENTERING FIPS_MODE" output via API on console.

In FIPS Approved mode, the module supports the following Approved Security Functions:

Table 4: Approved Security Functions

CAVP Cert.	Algorithm	Standard	Mode / Method	Key Lengths, Curves or Moduli	Use	Kernel Space or User Space
#4314	AES	FIPS 197, SP800-38A	CBC	128, 192, 256[1]	Data Encryption / Decryption	User Space
#4315	AES	FIPS 197, SP800-38A	CBC	128, 192, 256[1]	Data Encryption / Decryption	Kernel Space
#1028	CVL IKEv2 KDF[2]	SP800-135rev1	N/A	N/A	Key Derivation	User Space
#1027	CVL Partial DH	SP800-56A	FFC	2048	Key Pair Generation	User Space
#1372	DRBG	SP800-90A	CTR_DRBG[3], Hash_DRBG	N/A	Deterministic Random Bit Generation	User Space
#1149	DSA	FIPS 186-4	KeyPairGen	(2048, 224),[4] (2048, 256), (3072, 256)	Key Pair Generation; Prerequisite to KAS DH	User Space
#2846	HMAC	FIPS 197	HMAC-SHA-1, HMAC-SHA-256	112, 256	Message Authentication	User Space
#3549	SHS	FIPS 180-4	SHA-1, SHA-256, SHA-512[5]	N/A	Hashing	User Space

[1] Only AES-256 is utilized in the Approved mode of operation. All other key sizes are latent functionality and are not available in any service in the Approved or non-Approved mode of operation.
[2] It should be noted that no parts of the IKEv2 protocol, other than the KDF, have been tested by the CAVP.
[3] It should be noted that in the Approved mode of operation, the module only supports AES-256-CTR DRBG. All other modes are latent functionality.
[4] It should be noted that the only key size and mode utilized is (2048, 256). All other key size and modes are latent functionality and are not available in any service in the Approved or non-Approved mode of operation.
[5] It should be noted that SHA-512 is latent functionality and is not available in any service in the Approved or non-Approved mode of operation.

The following non-Approved algorithms and protocols are allowed within the Approved mode of operation:

Table 5: Non-Approved but allowed algorithms

Algorithm	Caveat	Use
Diffie-Hellman	Key agreement; key establishment methodology provides 112 bits of encryption strength	Key establishment within IKEv2 protocol
NDRNG	...	Seeding for the Approved DRBG

5.2 Non-Approved mode

The module provides the following non-FIPS approved algorithms only in non-FIPS mode of operation.

Table 6: Algorithms in Non-Approved mode

Non-Approved Algorithm	Usage / Description
AES CBC 128, 192, 256 (non-compliant)	Data Encryption / Decryption
CVL IKEv2 KDF (non-compliant)	Key Derivation
DRBG (non-compliant)	Deterministic Random Bit Generation
DSA (non-compliant)	Key Pair Generation; Prerequisite to KAS DH
HMAC-SHA-1 (non-compliant)	Message Authentication
HMAC-SHA-1-96 (non-compliant)	Message Authentication
IKEv2 KDF with HMAC-SHA-1 (non-compliant)	Key Derivation
KAS DH (non-compliant)	Key Pair Generation
SHA-1, SHA-256 (non-compliant)	Hashing
Triple-DES TCBC (non-compliant)	Data Encryption / Decryption

Table 7: Services available in non-Approved mode

Service	Usage / Description	Algorithms used
Configuration of PSK	Set up matching keywords for IPSec establishment	IKEv2 KDF
Establish IPSec tunnel	Perform DH key exchange and algorithm negotiation; Set up ipsec tunnel	AES HMAC IKEv2 KDF
Show tunnel status	Show status of IPSec tunnel	N/A
Display Module Status	Output status of module	N/A
Perform Self-Tests	Perform Self-Tests	N/A
Zeroization		KAS DH AES HMAC IKEv2 KDF AES DRBG

6 Identification and Authentication Policy

The Cryptographic Module supports an Administrator (Crypto Officer) role and a User role. A role is implicitly assumed based upon the service that is invoked.

Table 8: Roles and Required Identification and Authentication (FIPS 140-2 Table C1)

Role	Type of Authentication	Authentication Data
Administrator (Crypto Officer)	N/A	N/A
User	N/A	N/A

Table 9: Strengths of Authentication Mechanisms (FIPS 140-2 Table C2)

Authentication Mechanism	Strength of Mechanism
N/A	N/A

7 Access Control Policy

The following table describes the services of the module available in FIPS Approved Mode along with which role, cryptographic keys and CSPs, and type of access it corresponds to.

Table 10: Services Authorized for Roles, Access Rights within Services (FIPS 140-2 Table C3, Table C4)

Role	Service	Description	Cryptographic Keys & CSPs	Type(s) of Access R = Read W = Write D = Delete
Administrator	Installation of the module	Installation of the module	N/A	N/A
Administrator	Initialization of the module	Initialization of the module	N/A	N/A
Administrator	Configuration of PSK	Set up matching keywords for IPSec establishment	IKEv2 Pre-Shared Secret (PSK)	RWD
Administrator	Establish IPSec tunnel	Perform DH key exchange and algorithm negotiation; Set up IPSec tunnel	IKEv2 Pre-Shared Secret (PSK) IKEv2 AES Key IKEv2 HMAC Key IKEv2 KDF State IPSec AES Key IPSec HMAC Key	RWD
Administrator /User	Show tunnel status	Show status of IPSec tunnel	N/A	N/A
Administrator /User	Display Module Status	Output status of module	N/A	N/A
Administrator /User	Perform Self-Tests	Perform Self-Tests	N/A	N/A
Administrator /User	Zeroization		Local DH Private Key Local DH Public Key Peer DH Public Key DH Shared Secret IKEv2 Pre-Shared Secret (PSK) IKEv2 AES Key IKEv2 HMAC Key IKEv2 KDF State IPSec AES Key IPSec HMAC Key CTR_DRBG Internal State CTR_ DRBG Seed	W

7.1 Definition of Critical Security Parameters

The following list enumerates the secret keys, private keys, and public keys contained in the module. Details about the lifecycle of each cryptographic key and CSP can be found in Appendix A:

1. Local DH Private Key
2. Local DH Public Key
3. Peer DH Public Key
4. DH Shared Secret
5. IKEv2 Pre-Shared Secret (PSK)
6. IKEv2 AES Key
7. IKEv2 HMAC Key
8. IKEv2 KDF State
9. IPSec AES Key
10. IPSec HMAC Key
11. CTR_DRBG Internal State
12. CTR_DRBG Seed

8 Self-Tests

8.1 Power-Up Tests

1. Software Integrity Test:
    a. HMAC-SHA-1 (performed on all cryptographic module software (User Space and Kernel Space))
2. Known-Answer Tests:
    a. AES (256-bit) key size KAT (encrypt) in CBC Mode (Kernel Space and User Space)
    b. AES (256-bit) key size KAT (decrypt) in CBC Mode (Kernel Space and User Space)
    c. SHA-256 KAT (User Space Only)
    d. SP800-90A AES-256-CTR DRBG KAT (User Space Only)
    e. SP800-56A Diffie-Hellman KAT (User Space Only)
    f. SP800-135 IKEv2 KDF KAT (User Space Only)
    g. HMAC-SHA-256 KAT (User Space Only)
3. Critical Functions Tests:
    a. N/A

8.2 Conditional Tests

1. Bypass Test: N/A
2. Software Load Test: N/A
3. Continuous Random Number Generator (RNG) Test
    a. Performed on the output of the NDRNG (/dev/urandom)
    b. Performed on the output of the Approved SP800-90A CTR_DRBG

8.3 Self_test errors

1. In the event of a Software Integrity Test failure, the operator shall see the following message: "Fatpipe Integrity Check … different checksum [FAIL]"
2. In the event of a Power-On Self-Test failure in Kernel Space, the operator shall see the following error message: "Kernel Alg Test: [FAILED]"
3. In the event of a Power-On Self-Test failure in User Space, the operator shall see the following error message: "<self test name> failed"
4. In the event of a Conditional Self-Test failure for NDRNG and SP800-90A CTR_DRBG, the operator shall see the following error message: "OpenSSL internal error, assertion failed"

9 Physical Security Policy

The Fatpipe Crypto Module is a software module. The physical security requirements are not applicable.

Table 11: Inspection/Testing of Physical Security Mechanisms

Physical Security Mechanisms	Recommended Frequency of Inspection/Test	Inspection/Test Guidance Details
N/A	N/A	N/A

10 Mitigation of Other Attacks Policy

The Fatpipe Crypto Module is not designed to mitigate any specific attacks.

Table 12: Mitigation of Other Attacks (FIPS 140-2 Table C6)

Other Attacks	Mitigation Mechanism	Specific Limitations
N/A	N/A	N/A

11 Glossary

The following table defines the acronyms used in this document.

Table 13: Acronym Table

Term/Acronym	Description
AES	Advanced Encryption Standard
CBC	Cipher-Block Chaining
CSP	Critical Security Parameter
CVL	Component Validation List
DES	Data Encryption Standard
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Message Authentication Code
IKEv2	Internet Key Exchange
IPSec	Internet Protocol Security
KAS	Key Agreement Scheme
KAT	Known-Answer Test
KDF	Key Derivation Function
NDRNG	Non-deterministic Random Number Generator
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard

12 References

Title	Link
Security Requirements for Cryptographic Modules (FIPS PUB 140-2)	http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf
Advance Encryption Standard (FIPS PUB 197)	http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf
Recommendation for Random Number Generation Using Deterministic Random Bit Generators (NIST SP 800-90A)	http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
Digital Signature Standard (FIPS PUB 186-4)	http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
The Keyed-Hash Message Authentication Code (FIPS PUB 198-1)	http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.198-1.pdf
Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (NIST SP 800-56A)	http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf
Secure Hash Standard (FIPS PUB 180-4)	http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

Appendix A: Critical Security Parameters and Public Keys

1. Local DH Private Key
   Type: minimum 2048-bit
   Generation: SP800-90A. As per SP800-133 Section 6.2, the random value (K) needed to generate key pairs for the finite field is the output of the SP800-90 DRBG; this is Approved as per SP800-56A.
   Establishment: N/A
   Entry: N/A
   Output: N/A
   Storage: plaintext in RAM (fresh for each session)
   Zeroization: by power cycle

2. Local DH Public Key
   Type: minimum 2048-bit
   Generation: SP800-90A. As per SP800-133 Section 6.2, the random value (K) needed to generate key pairs for the finite field is the output of the SP800-90 DRBG; this is Approved as per SP800-56A.
   Establishment: N/A
   Entry: N/A
   Output: signed HMAC with PSK
   authenticated by HMAC-SHA1
   Storage: plaintext in RAM (fresh for each session)
   Zeroization: by power cycle

3. Peer DH Public Key
   Type: minimum 2048-bit
   Generation: N/A
   Establishment: N/A
   Entry: signed with HMAC with PSK
   authenticated by HMAC-SHA1
   Output: N/A
   Storage: plaintext in RAM
   Zeroization: by power cycle

4. DH Shared Secret
   Type: minimum 2048-bit
   Generation: N/A
   Establishment: via IKEv2 negotiation
   Entry: N/A
   Output: N/A
   Storage: plaintext in RAM
   Zeroization: by power cycle

5. IKEv2 Pre-Shared Secret (PSK)
   Type: 112-bit HMAC key used for authentication during IKE negotiations
   Generation: N/A
   Establishment: N/A
   Entry: N/A as per FIPS 140-2 IG 7.7
   Output: N/A

Storage: Plaintext
Zeroization: Active overwrite

6. IKEv2 AES Key
   Type: 256-bit key. Used in CBC mode to encrypt/decrypt within IKEv2.
   Generation: SP800-135 KDF
   Establishment: via IKEv2 negotiation
   Entry: N/A
   Output: N/A
   Storage: plaintext in RAM
   Zeroization: by power cycle

7. IKEv2 HMAC Key
   Type: HMAC-SHA-1 key (at least 112 bits)
   Generation: N/A
   Establishment: via IKEv2 negotiation
   Entry: N/A
   Output: N/A
   Storage: plaintext in RAM
   Zeroization: by power cycle

8. IKEv2 KDF State
   Type: HMAC-SHA-256
   Generation: N/A
   Establishment: via IKEv2 negotiation
   Entry: N/A
   Output: N/A
   Storage: plaintext in RAM
   Zeroization: by power cycle

9. IPSec AES Key
   Type: 256-bit key. Used in CBC mode to encrypt/decrypt within IPSec.
   Generation: N/A
   Establishment: via IKEv2 negotiation
   Entry: N/A
   Output: N/A
   Storage: plaintext in RAM
   Zeroization: by power cycle

10. IPSec HMAC Key
    Type: HMAC-SHA-1 key (at least 112 bits)
    Generation: N/A
    Establishment: via IKEv2 negotiation
    Entry: N/A
    Output: N/A
    Storage: plaintext in RAM
    Zeroization: by power cycle

11. CTR_DRBG Internal State

Type: SP800-90A CTR_DRBG
Generation: seeded by /dev/urandom
Establishment: N/A
Entry: N/A
Output: N/A
Storage: plaintext in RAM
Zeroization: by power cycle

12. CTR_DRBG Seed
    Type: SP800-90A CTR_DRBG
    Generation: seeded by /dev/urandom
    Establishment: N/A
    Entry: N/A
    Output: N/A
    Storage: plaintext in RAM
    Zeroization: by power cycle

Appendix B: CKG as per SP800-133

In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) as per SP800-133 (vendor affirmed). The resulting generated seed, for asymmetric key generation, is the unmodified output from the SP800-90A DRBG. Please see Appendix A: Critical Security Parameters and Public Keys for more information.