FIPS 140-2 Non-Proprietary Security Policy

FortiGate-VM 6.4 and 7.0

FortiGate-VM 6.4 and 7.0 FIPS 140-2 Non-Proprietary Security Policy	
Document Version:	1.9
Publication Date:	Thursday, December 15, 2022
Description:	Documents FIPS 140-2 Level 1 Security Policy issues, compliancy and requirements for FIPS compliant operation.
Software Versions:	FortiOS 6.4 (FIPS-CC-64-5) FortiOS 7.0 (FIPS-CC-70-6)
Hardware Versions:	Intel® Xeon® D-1559, Intel® Xeon® E3-1515M and Intel® Xeon® E-2276ME

FORTINET DOCUMENT LIBRARY

https://docs.fortinet.com

FORTINET VIDEO GUIDE

https://video.fortinet.com

FORTINET KNOWLEDGE BASE

http://kb.fortinet.com

FORTINET BLOG

https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT

https://www.fortinet.com/support/contact.html

FORTINET NSE INSTITUTE (TRAINING)

https://training.fortinet.com/

FORTIGUARD CENTER

https://fortiguard.com

FORTICAST

http://forticast.fortinet.com

END USER LICENSE AGREEMENT AND PRIVACY POLICY

https://www.fortinet.com/doc/legal/EULA.pdf

https://www.fortinet.com/corporate/about-us/privacy.html

FEEDBACK

Email: techdoc@fortinet.com

Thursday, December 15, 2022

FortiGate-VM 6.4 and 7.0 FIPS 140-2 Non-Proprietary Security Policy

01-648-787234-20220302

TABLE OF CONTENTS

Overview

This document is a FIPS 140-2 Security Policy for Fortinet Incorporated's FortiGate-VM Next Generation Firewall virtual appliances running FortiOS 6.4 and 7.0. This policy describes how the appliances (hereafter referred to as the 'module') meet the FIPS 140-2 security requirements and how to operate the module in a FIPS compliant manner. This policy was created as part of the FIPS 140-2 Level 1 validation of the module.

The Federal Information Processing Standards Publication 140-2 - *Security Requirements for Cryptographic Modules* (FIPS 140-2) details the United States Federal Government requirements for cryptographic modules. Detailed information about the FIPS 140-2 standard and validation program is available on the NIST (National Institute of Standards and Technology) website at http://csrc.nist.gov/groups/STM/cmvp/index.html.

References

This policy deals specifically with operation and implementation of the modules in the technical terms of the FIPS 140-2 standard and the associated validation program. Other Fortinet product manuals, guides and technical notes can be found at the Fortinet technical documentation website at https://docs.fortinet.com.

Additional information on the entire Fortinet product line can be obtained from the following sources:

- Find general product information in the product section of the Fortinet corporate website at https://www.fortinet.com/products.
- Find on-line product support for registered products in the technical support section of the Fortinet corporate website at https://www.fortinet.com/support.
- Find contact information for technical or sales related questions in the contacts section of the Fortinet corporate website at https://www.fortinet.com/contact.
- Find security information and bulletins in the FortiGuard Center of the Fortinet corporate website at https://www.fortiguard.com.

Introduction

FortiGate Virtual Appliances allow you to mitigate blind spots by implementing critical security controls within your virtual infrastructure. They also allow you to rapidly provision security infrastructure whenever and wherever it is needed. FortiGate virtual appliances feature all of the security and networking services common to traditional hardware-based FortiGate appliances

FortiGate virtual appliances detect and eliminate the most damaging, content-based threats from email and Web traffic such as viruses, worms, intrusions, inappropriate Web content and more in real time — without degrading network performance. In addition to providing application level firewall protection, FortiGate virtual appliances deliver a full range of network-level services including VPN, intrusion prevention, web filtering, antivirus, antispam and traffic shaping.

FortiGate virtual appliances support the IPsec industry standard for VPN, allowing VPNs to be configured between a FortiGate virtual appliance and any client or gateway/firewall that supports IPsec VPN. FortiGate virtual appliances also provide SSL VPN services using TLS 1.1 and 1.2.

Security Level Summary

The module meets the overall requirements for a FIPS 140-2 Level 1 validation.

Table 1: Summary of FIPS security requirements and compliance levels

Security Requirement	Compliance Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services and Authentication	3
Finite State Model	1
Physical Security	1
Operational Environment	n/a
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	2
Mitigation of Other Attacks	1

Module Description

The FortiGate-VM virtual appliances are software-hybrid security modules designed to execute on a general purpose computer (GPC) hardware platform. The module is a multi-chip standalone cryptographic module. As a software-hybrid cryptographic module, the only physical characteristic of the virtual appliance is the use of the Intel® CPU which provides “Advanced Encryption Standard New Instructions (AES-NI)” when running on the host GPC. The module must rely on physical characteristics of the host system on which it runs. The module supports the physical interfaces of the hardware it is running on. See Figure 1 for a block diagram of the physical system. The module utilizes physical interfaces of the tested platform hosting the virtual environment upon which the module is installed. The hypervisor running on the physical system controls and maps the module's virtual interfaces to the physical interfaces, which include the CPU, memory, network interfaces and hard disk. The module binary is distributed as FortiGate-VM64.ovf and fortios.vmdk for deployment in VMware ESXi. The only hardware component within the cryptographic boundary is the CPU, which is a production grade component with standard passivation.

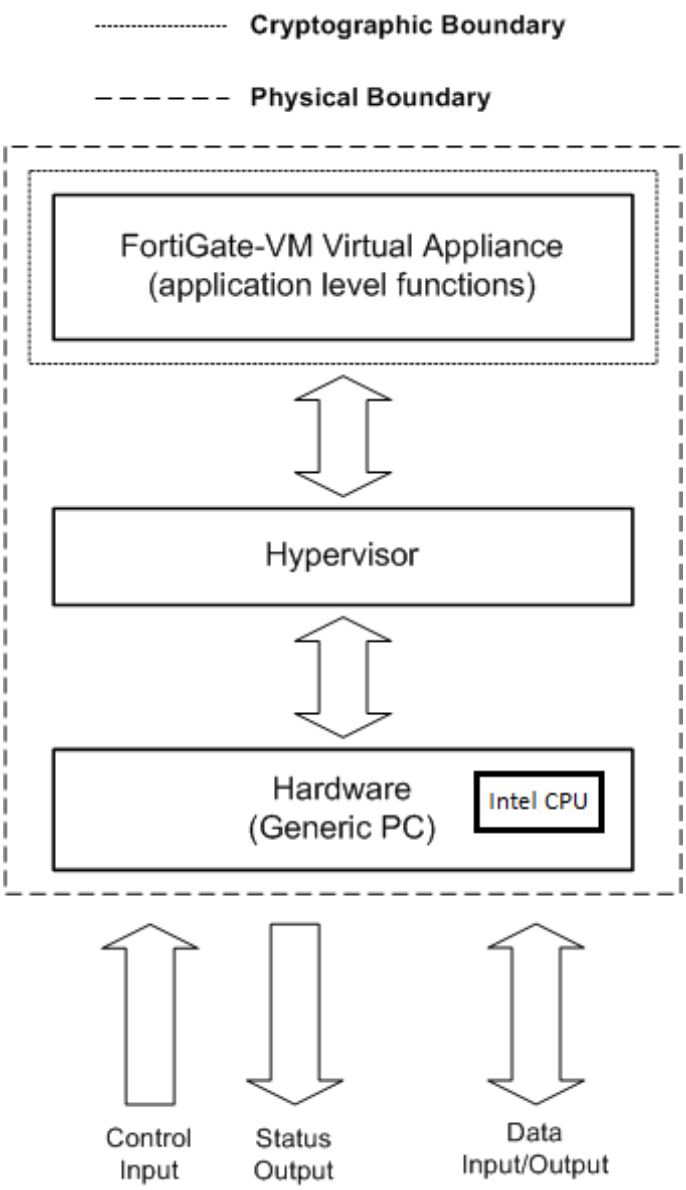

Figure 1: FortiGate-VM Physical and Cryptographic Boundaries

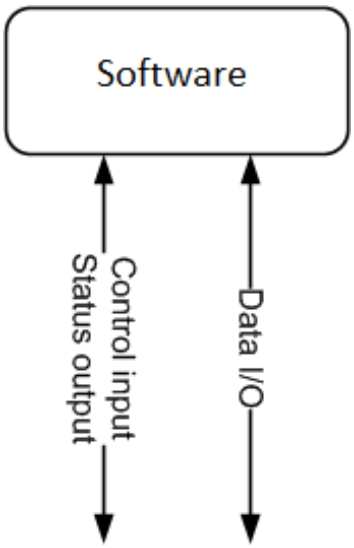

Figure 2: FortiOS logical cryptographic boundary

Figure 3: Intel Processors Xeon D-1559, Xeon E3-1515M, Xeon E-2276ME (Hardware Components)

The validated software versions are FortiOS 6.4 (FIPS-CC-64-5) and FortiOS 7.0 (FIPS-CC-70-6). Any software version that is not shown on the module certificate is out of the scope of this validation and requires a separate FIPS 140-2 validation

Note that no claim can be made 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.

Module Interfaces

The module's logical interfaces and physical ports are described in the table below.

Table 2: FortiGate-VM logical interfaces and physical ports

FIPS 140 Interface	FortiGate-VM Interface	Logical Interface	Physical Interface
Data Input	Virtual Ethernet Ports, Virtual USB Ports	API input parameters	Network interface, USB interface (Entropy Token)
Data Output	Virtual Ethernet Ports	API output parameters	Network Interface
Control Input	Virtual Ethernet Ports, Virtual Serial Ports, Virtual USB Ports	API function calls	Network Interface, serial interface, USB interface (USB token)
Status Output	Virtual Ethernet Ports, Virtual Serial Ports	API return values	Network interface, serial interface
Power Input	N/A	N/A	The power supply is the power interface

Web-Based Manager

The FortiGate web-based manager provides GUI based access to the modules and is the primary tool for configuring the modules. The manager requires a web browser on the management computer and an Ethernet connection between the FortiGate unit and the management computer.

A web-browser that supports Transport Layer Security 1.2 (or TLS 1.1 if permitted) is required for remote access to the web-based manager when the module is operating in FIPS-CC mode. HTTP access to the web-based manager is not allowed in FIPS-CC mode and is disabled.

Command Line Interface

The FortiGate Command Line Interface (CLI) is a full-featured, text based management tool for the module. The CLI provides access to all of the possible services and configuration options in the module. The CLI uses a console connection or a network (Ethernet) connection between the FortiGate unit and the management computer. The console connection is a direct serial connection. Terminal emulation software is required on the management computer using either method. For network access, a Telnet or SSH client that supports the SSH v2.0 protocol is required (SSH v1.0 is not supported in FIPS-CC mode). Telnet access to the CLI is not allowed in FIPS-CC mode and is disabled.

Roles, Services and Authentication

Roles

When configured in FIPS-CC mode, the module provides the following roles:

- Crypto Officer
- Network User

The Crypto Officer role is initially assigned to the default ‘admin’ operator account. The Crypto Officer role has read-write-execute access to all of the module’s administrative services. The initial Crypto Officer can create additional operator accounts. These additional accounts are assigned the Crypto Officer role and can be assigned a range of read-write-execute or read only access permissions including the ability to create operator accounts.

The modules also provide a Network User role for end-users (Users). Network Users can make use of the encrypt/decrypt services, but cannot access the modules for administrative purposes.

The module does not provide a Maintenance role.

FIPS Approved Services

The following tables detail the types of FIPS approved services available to each role in each mode of operation, the types of access for each role and the Keys or CSPs they affect.

The access types are abbreviated as follows:

Read Access	R
Write Access	W
Execute Access	E

Table 3: Services available to Crypto Officers

Service	Access	Key/CSP
connect to module using the virtual console port	WE	N/A
connect to module remotely using TLS*	WE	Diffie-Hellman Keys, EC Diffie Hellman Keys,TLS Premaster Secret, TLS Master Secret, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Integrity Key, and HTTPS/TLS Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, Entropy String, TLS Server Signatures
connect to module remotely using SSH*	WE	Diffie-Hellman Keys, SSH Server/Host Key, SSH Session Authentication Key, SSH Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, Entropy String
authenticate to module	WE	Crypto Officer Password
show system status	N/A	N/A
show FIPS-CC mode enabled/disabled (console/CLI only)	N/A	N/A

Service	Access	Key/CSP
enable FIPS-CC mode of operation (console only)	WE	Configuration Integrity Key
key zeroization	W	All Keys
execute factory reset (disable FIPS-CC mode, console/CLI only)	W	N/A
execute FIPS-CC on-demand self-tests (console only)	E	Configuration Integrity Key, Software Integrity Key
add/delete crypto officers and network users	WE	Crypto Officer Password, Network User Password
set/reset crypto officers and network user passwords	WE	Crypto Officer Password, Network User Password
backup/restore configuration file	RWE	Configuration Encryption Key, Configuration Backup Key
read/set/delete/modify module configuration*	N/A	N/A
execute software update	WE	Software Update Key
read log data	N/A	N/A
delete log data (console/CLI only)	N/A	N/A
execute system diagnostics (console/CLI only)	N/A	N/A
enable/disable alternating bypass mode	N/A	N/A
read/set/delete/modify IPsec/SSL VPN configuration*	W	IPsec: IPsec Manual Authentication Key, IPsec Manual Encryption Key, IKE Pre-Shared Key, IKE RSA Key, IKE ECDSA Key, Diffie-Hellman Keys, EC Diffie-Hellman Keys SSL: HTTPS/TLS Server/Host Key, HTTPS/TLS Session Integrity Key, HTTPS/TLS Session Encryption Key
read/set/modify HA configuration	WE	HA Password, HA Encryption Key
log offloading to remote FortiAnalyzer device*	E	OFTP Client Key, Diffie-Hellman Keys, EC Diffie-Hellman Keys, TLS Premaster Secret, TLS Master Secret, HTTPS/TLS Session Integrity Key, HTTPS/TLS Session Encryption Key, HTTPS/TLS Server Host Key, DRBG v and key values, DRBG Output, DRBG Seed, Entropy String
generate CSR with RSA or ECDSA	WE	RSA keys, ECDSA keys
Intel CPU - PAA that performs cryptographic acceleration	N/A	N/A

Table 4: Services available to Network Users in FIPS-CC mode

Service/CSP	Access	Key/CSP
connect to module remotely using TLS*	WE	Diffie-Hellman Keys, EC Diffie-Hellman Keys, TLS Premaster Secret, TLS Master Secret, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Integrity Key, HTTPS/TLS Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, Entropy String, TLS Server Signatures
authenticate to module	WE	Network User Password
IPsec VPN controlled by firewall policies*	E	IPsec: IPsec Manual Authentication Key, IPsec Manual Encryption Key, IPsec Session Authentication Key, IPsec Session Encryption Key, IKE Pre-Shared Key, IKE RSA Key, IKE ECDSA Key, IKE SKEYSEED, IKE Authentication Key, IKE Key Generation Key, IKE Session Encryption Key, Diffie-Hellman Keys, EC Diffie-Hellman Keys
SSL VPN controlled by firewall policies*	E	Network User Password, Diffie-Hellman Keys, EC Diffie-Hellman Keys, TLS Premaster Secret, TLS Master Secret, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Integrity Key, HTTPS/TLS Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, Entropy String
Intel CPU - PAA that performs cryptographic acceleration	N/A	N/A

Non-FIPS Approved Services

The module also provides the following non-FIPS approved services:

- Configuration backups using password protection
- L2TP and PPTP VPN
- Services marked with an asterisk (*) in Tables 4 and 5 are considered non-approved when using the following algorithms:
  - Non-compliant-strength Diffie-Hellman

The above services shall not be used in the FIPS approved mode of operation.

Authentication

The module implements identity based authentication. Crypto Officers must authenticate with a user-id and password combination to access the modules remotely or locally via the console. Remote Crypto Officer authentication is done over HTTPS (TLS) or SSH. The password entry feedback mechanism does not provide information that could be used to guess or determine the authentication data.

Authentication at level 3 is only applicable when identity-based authentication is enforced for the User role.

By default, Network User access to the modules is based on firewall policy and authentication by IP address or fully qualified domain names. Network Users can optionally be forced to authenticate to the modules using a username/password combination to enable use of the IPsec VPN encrypt/decrypt or bypass services. For Network Users invoking the SSL-VPN encrypt/decrypt services, the modules support authentication with a user-id/password combination. Network User authentication is done over HTTPS and does not allow access to the modules for administrative purposes.

The minimum password length is 8 characters when in FIPS-CC mode (maximum password length is 128 characters) chosen from the set of ninety four (94) characters. New passwords are required to include 1 uppercase character, 1 lowercase character, 1 numeric character, and 1 special character. The odds of guessing a password are 1 in 3,346,172,314,938,369 which is significantly lower than one in a million.

Note that Crypto Officer authentication over HTTPS/SSH and Network User authentication over HTTPS are subject to a limit of 3 failed authentication attempts in 1 minute; thus, the maximum number of attempts in one minute is 3. Therefore the probability of a success with multiple consecutive attempts in a one-minute period is 3 in 3,346,172,314,938,369 which is less than 1/100,000.

Crypto Officer authentication using the console is not subject to a failed authentication limit, but the number of authentication attempts per minute is limited by the bandwidth available over the serial connection which is a maximum of 115,200 bps which is 6,912,000 bits per minute. An 8 byte password would have 64 bits, so there would be no more than 108,000 passwords attempts per minute. Therefore the probability of success would be 1/({3,346,172,314,938,369}/108,000) which is less than 1/100,000.

For Network Users invoking the IPsec VPN encrypt/decrypt services, the module acts on behalf of the Network User and negotiates a VPN connection with a remote module. The strength of authentication for IPsec services is based on the authentication method defined in the specific firewall policy: IPsec manual authentication key, IKE pre-shared key, IKE RSA key (RSA certificate) or IKE ECDSA key (ECDSA certificate). The odds of guessing the authentication key for each IPsec method is:

- 1 in 16^40 for the IPsec Manual Authentication key (based on a 40 digit, hexadecimal key)
- 1 in 94^8 for the IKE Pre-shared Key (based on an 8 character, ASCII printable key)
- 1 in 2^112 for the IKE RSA Key (based on a 2048 bit RSA key size)
- 1 in 2^128 for the IKE ECDSA Key (based on a P-256 curve ECDSA key size)

A gigabit ethernet connection is 1,048,576,000 bits per second which is 62,914,560,000 bits per minute. An 8-byte key would have 64 bits, so there could be no more than 983,040,000 password attempts per minute. Therefore, the minimum odds of guessing the IKE Preshared key for IPSec within a one-minute period is 1 in 94^8/983,040,000 which is less than 1 in 100,000. Similarly, for the IPsec Manual Authentication key, the minimum odds of Network Users guessing the key within a minute would be 1 in 16^40/393,216,000. Guessing the IKE RSA key within a minute would be 1 in 2^112/561,737,143. Guessing the IKE ECDSA key within a minute would be 1 in 2^128/491,520,000.

Operational Environment

The operational environment for the module consists of the FortiGate-VM software and VMware ESXi 6.7 hypervisor running on the appliances listed below. The underlying OS is FortiGate-VM 6.4 or 7.0. No claim can be made 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.

The module can also be executed on any PacStar 451 or PacStar 455 appliance and be vendor affirmed FIPS-compliant.

Appliance	Platform	OS	CPU
PacStar 451 (P/N: 075-0451-165)	VMWare ESXi 6.7	FortiOS 6.4/7.0	Intel® Xeon® E-2276ME
PacStar 451 (P/N: 075-0451-55)	VMWare ESXi 6.7	FortiOS 6.4/7.0	Intel® Xeon® D-1559
PacStar 451 (P/N: 075-0451-45)	VMWare ESXi 6.7	FortiOS 6.4/7.0	Intel® Xeon® E3-1515M

In addition, the module is vendor affirmed on the following cloud platforms with version FortiOS 7.0 (FIPS-CC-70-6):

Cloud Platform
Amazon AWS
Microsoft Azure

Cryptographic Key Management

Random Number Generation

The modules use a software based, deterministic random bit generator (DRBG) that conforms to NIST Special Publication 800-90A. The Module generates cryptographic keys whose strengths are modified by available entropy.

Entropy

The modules use an entropy token (Araneus Alea II) to seed the DRBG during the modules' boot process and to periodically reseed the DRBG. The entropy token is not included in the boundary of the module and therefore no assurance can be made for the correct operation of the entropy token nor is there a guarantee of stated entropy.

Entropy Strength

The entropy loaded into the approved AES-256 bit DRBG is 256 bits. The entropy source is over-seeded and then an HMAC-SHA-256 post-conditioning component (as per section 3.1.5 of SP 800-90B) is applied.

Reseed Period

The RBG is seeded from the Entropy Token during the boot process and then reseeded periodically. The default reseed period is once every 24 hours (1440 minutes) and is configurable (1 to 1440 minutes). The entropy token must be installed to complete the boot process and to reseed the main PCB DRBG instance.

Key Zeroization

The zeroization process must be performed under the direct control of the operator. The operator must be present to observe that the zeroization method has completed successfully.

All keys and CSPs are zeroized by the following procedure:

1. At the console CLI, log in as admin and issue the "execute factoryreset" command.
2. After waiting for a restart, log in as admin via the GUI, go to System, and update the software by loading a new ".out" software file.

Algorithms

Table 5: FIPS approved algorithms

Algorithm	NIST Cert Number
CTR DRBG (NIST SP 800-90A) with AES 256 bits	A2291, A2292
AES in CBC mode (128, 192, and 256 bits)	A2291, A2298, A2292, A2299
AES in GCM mode (128, 256 bits)	A2291, A2298, A2292, A2299
SHA-1	A2291, A2298, A2292, A2299
SHA-224	A2298, A2299
SHA-256	A2298, A2299
SHA-384	A2298, A2299
SHA-512	A2298, A2299
HMAC SHA-1	A2291, A2298, A2292, A2299
HMAC-SHA-224	A2298, A2299
HMAC SHA-256	A2298, A2299
HMAC SHA-384	A2298, A2299
HMAC SHA-512	A2298, A2299
RSA PKCS 1.5 Key Pair Generation: 2048 and 3072-bit Signature Generation: 2048 and 3072-bit Signature Verification: 1024, 2048 and 3072-bit For legacy use, the module supports 1024-bit RSA keys and SHA-1 for signature verification	A2298, A2299

Algorithm	NIST Cert Number
RSA PSS Signature Generation: 2048 3072, and 4096-bit Signature Verification: 1024, 2048, 3072 and 4096-bit	A2298, A2299
ECDSA Key Pair Generation: curve P-256	A2298, A2299
ECDSA Key Pair Generation: curve P-384	A2298, A2299
ECDSA Key Pair Generation: curve P-521	A2298, A2299
ECDSA Signature Generation: curves P-256, P-384 and P-521	A2298, A2299
ECDSA Signature Verification: curves P-256, P-384 and P-521	A2298, A2299
CVL (KDF SSH) - AES 128 bit-, AES-192 bit, AES 256 bit - CBC (using SHA1, SHA-256)	A2298, A2299
CVL (KDF TLS 1.2 RFC7627(using SHA-256, SHA-384,SHA-512))	A2298, A2299
CVL (KDF IKE v1 (using SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512))	A2298, A2299
CVL (KDF IKE v2 (using SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512))	A2298, A2299
KAS-ECC-SSC SP800-56Ar3	A2298, A2299
KAS-FFC-SSC SP800-56Ar3	A2298, A2299
CVL (KDF SNMP) - Password length: 64 - 8192	A2298, A2299

KTS (AES Certs. #A2298 and #A2299 [128, 256-bit AES-CBC] and HMAC Certs. #A2298 and #A2299; key establishment methodology provides 128 or 256 bits of encryption strength);

KTS (AES Certs. #A2298 and #A2299 [128, 256-bit AES-GCM]; key establishment methodology provides 128 or 256 bits of encryption strength);

KAS-ECC-SSC provides between 128 and 256 bits of encryption strength;

KAS-FFC-SSC provides 112 bits of encryption strength;

KAS (KAS-SSC Certs. #A2298 and #A2299, CVL Certs. #A2298 and #A2299)

For AES GCM IPsec/IKEv2, RFC 7296 is used to establish the shared secret SKEYSEED from which the AES GCM encryption keys are derived.

Table 6: Non-FIPS approved algorithms. Not Allowed.

Algorithm
Diffie-Hellman is non-compliant when keys less than 2048 bits are used, since such keys do not provide the minimum required 112 bits of encryption strength.
The module implements the following non-NIST recommended curves: Curve25519 and Curve448. Until such time as NIST SP 800-186 is published, these curves remain non-recommended by NIST. The module may employ these curves for TLS interoperability; however, it is the responsibility of the operator to utilize cipher suites which contain only NIST Approved cryptography.

Note that the IKE, SSH, SNMP, and TLS protocols, other than the KDF, have not been tested by the CMVP or CAVP as per FIPS 140-2 Implementation Guidance D.11.

The module is compliant to IG A.5: GCM is used in the context of TLS and IKEv2/IPSec.

For TLS, The GCM implementation meets Option 1 of IG A.5: it is used in a manner compliant with SP 800-52 and in accordance with RFC 5246 for TLS key establishment. The AES GCM IV generation is in compliance with RFC 5288 and shall only be used for the TLS protocol version 1.2 to be compliant with FIPS140-2 IG A.5, Option 1 (“TLS protocol IV generation”); thus, those cipher suites implemented in the module that utilize AES-GCM are consistent with those specified in Section 3.3.1.1.2 of [SP800-52, Rev2]. During operational testing, the module was tested against an independent version of TLS and found to behave correctly.

For IPsec/IKEv2, the GCM implementation meets Option 1 of IG A.5: it is used in a manner compliant with RFCs 4106 and 7296. During operational testing, the module was tested against an independent version of IPsec with IKEv2 and found to behave correctly.

For SSH, the module is compliant with RFC 4252, 4253, and 5647.

In case the module’s power is lost and then restored, the key used for the AES GCM encryption or decryption shall be re-distributed. In addition, when the nonce_explicit part of the IV exhausts the maximum number of values for a session key a handshake is triggered to establish a new encryption key.

There are algorithms, modes, and keys that have been CAVs tested but are not used by the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in the above tables are used by the module.

Note that the TLS KDF has only been CAVP tested for TLS 1.2 and not TLS 1.1. As such, TLS 1.1 should not be used in the approved mode of operation.

Cryptographic Keys and Critical Security Parameters

The following table lists all of the cryptographic keys and critical security parameters used by the modules. The following definitions apply to the table.

Table 7: Cryptographic Keys and Critical Security Parameters used in FIPS-CC mode

Key or CSP	Generation	Storage	Usage	Zeroization
Entropy string	Entropy Token	Boot device Plain-text	Input string for the entropy pool	By execute 'factoryreset' command / apply software
DRBG seed	Internally generated	Boot device Plain-text	256 bit seed used by the DRBG (output from entropy token)	By execute 'factoryreset' command / apply software
DRBG output	Internally generated	Boot device Plain-text	Random numbers used in cryptographic algorithms (256 bits)	By execute 'factoryreset' command / apply software
DRBG v and key values	Internally generated	Boot device Plain-text	Internal state values for the CTR_DRBG 256	By execute 'factoryreset' command / apply software
IPsec Manual Authentication Key	Electronic key entry	Boot device AES encrypted	Used as IPsec Session Authentication Key	By execute 'factoryreset' command / apply software
IPsec Manual Encryption Key	Electronic key entry	SDRAM Plain-text	Used as IPsec Session Encryption Key using AES (128, 256 bit)	By execute 'factoryreset' command / apply software
IPsec Session Authentication Key	Internally generated using DRBG	SDRAM Plain-text	IPsec peer-to-peer authentication using HMAC SHA-1 or HMAC SHA-256	By execute 'factoryreset' command / apply software
IPsec Session Encryption Key	Internally generated via DH or ECDH KAS	SDRAM Plain-text	VPN traffic encryption/decryption using AES (128, 256 bit)	By execute 'factoryreset' command / apply software
IKE SKEYSEED	Derived via KDF defined in SP800-135 (IKEv2)	SDRAM Plain-text	Used to generate IKE protocol keys	By execute 'factoryreset' command / apply software

Key or CSP	Generation	Storage	Usage	Zeroization
IKE Pre-Shared Key	Electronic key entry	Boot device AES encrypted	Used to generate IKE protocol keys	By execute 'factoryreset' command / apply software
IKE Authentication Key	Internally generated using DRBG	SDRAM Plain-text	IKE peer-to-peer authentication using HMAC SHA-1 , -256, -384 or -512	By execute 'factoryreset' command / apply software
IKE Key Generation Key	Internally generated using DRBG	SDRAM Plain-text	IPsec SA keying material	By execute 'factoryreset' command / apply software
IKE Session Encryption Key	Internally generated via DH or ECDH KAS	SDRAM Plain-text	Encryption of IKE peer-to-peer key negotiation using or AES (128, 256 bit)	By execute 'factoryreset' command / apply software
IKE RSA Key	Externally generated	Boot device Plain-text	RSA private key used in the IKE protocol (2048 and 3072 bit signatures)	By execute 'factoryreset' command / apply software
IKE ECDSA Key	Externally generated	Boot device Plain-text	ECDSA private key used in the IKE protocol (signatures using P-256, P-384 and P-521 curves)	By execute 'factoryreset' command / apply software
Diffie-Hellman Keys	Internally generated using DRBG	SDRAM Plain-text	Key agreement and key establishment (Public key size of 2048 to 8192 bits with Private key size of 224 to 400 bits)	By execute 'factoryreset' command / apply software
EC Diffie-Hellman Keys	Internally generated using DRBG	SDRAM Plain-text	Key agreement and key establishment (key pairs on the curves secp256r1, secp384r1 and secp521r1)	By execute 'factoryreset' command / apply software

Key or CSP	Generation	Storage	Usage	Zeroization
Software Update Key	Preconfigured	Boot device Plain-text	Verification of software integrity when updating to new software versions using RSA public key (software load test, 2048-bit signature)	By execute 'factoryreset' command / apply software
Software Integrity Key	Preconfigured	Boot device Plain-text	Verification of software integrity in the software integrity test using RSA public key (software integrity test, 2048 bit signature)	By execute 'factoryreset' command / apply software
TLS Premaster Secret	Internally generated via DH or ECDH KAS	SDRAM Plain-text	HTTPS/TLS keying material	By execute 'factoryreset' command / apply software
TLS Master Secret	Internally generated from the TLS Premaster Secret	SDRAM Plain-text	384 bit master key used in the HTTPS/TLS protocols	By execute 'factoryreset' command / apply software
HTTPS/TLS Server/Host Key	Preconfigured	Boot device Plain-text	RSA private key used in the HTTPS/TLS protocols (key establishment, 2048 or 3072 bit)	By execute 'factoryreset' command / apply software
HTTPS/TLS Session Integrity Key	Internally generated using DRBG	SDRAM Plain-text	HMAC SHA-1, -256 or -384 key used for HTTPS/TLS session integrity	By execute 'factoryreset' command / apply software
TLS Server Signatures	Preconfigured	Boot device Plain-text	rsa_pkcs1 & rsa_pss_ rsae signatures used in TLS	By execute 'factoryreset' command / apply software
HTTPS/TLS Session Encryption Key	Internally generated via DH or ECDH KAS	SDRAM Plain-text	AES (128, 256 bit) key used for HTTPS/TLS session encryption	By execute 'factoryreset' command / apply software

Key or CSP	Generation	Storage	Usage	Zeroization
SSH Server/Host Key	Preconfigured	Boot device Plain-text	RSA private key used in the SSH protocol (key establishment, 2048 or 3072 bit)	By execute 'factoryreset' command / apply software
SSH Session Authentication Key	Internally generated using DRBG	SDRAM Plain-text	HMAC SHA-1 or HMAC SHA-256 key used for SSH session authentication	By execute 'factoryreset' command / apply software
SSH Session Encryption Key	Internally generated via DH or ECDH KAS	SDRAM Plain-text	AES (128, 256 bit) key used for SSH session encryption	By execute 'factoryreset' command / apply software
Crypto Officer Password	Electronic key entry	Boot device SHA-1 hash	Used to authenticate operator access to the module	By execute 'factoryreset' command / apply software
Configuration Integrity Key	Preconfigured	Boot device Plain-text	HMAC SHA-256 hash used for configuration bypass test	By execute 'factoryreset' command / apply software
Configuration Encryption Key	Preconfigured	Boot device Plain-text	AES 256 bit key used to encrypt CSPs on the Boot device and in the backup configuration file (except for crypto officer passwords in the backup configuration file)	By execute 'factoryreset' command / apply software
Configuration Backup Key	Preconfigured	Boot device Plain-text	HMAC SHA-256 key used to hash crypto officer passwords in the backup configuration file	By execute 'factoryreset' command / apply software
Network User Password	Electronic key entry	Boot device SHA-1 hash	Used to authenticate network access to the module	By execute 'factoryreset' command / apply software
HA Password	Electronic key entry	Boot device AES encrypted	Used to authenticate FortiGate units in an HA cluster	By execute 'factoryreset' command / apply software

Key or CSP	Generation	Storage	Usage	Zeroization
HA Encryption Key	Externally generated	Boot device AES encrypted	Encryption of traffic between units in an HA cluster using AES 128 bit key	By execute 'factoryreset' command / apply software
OFTP Client Key	Externally generated	Boot device AES encrypted	RSA private key used in the OFTP/TLS protocol (key establishment, 2048 bit signature)	By execute 'factoryreset' command / apply software
RSA Keys	Internally generated using DRBG	Boot device Plain-text	RSA Key Pair from RSA CSR generation	By execute 'factoryreset' command / apply software
ECDSA Keys	Internally generated using DRBG	Boot device Plain-text	ECDSA Key Pair from ECDSA CSR generation	By execute 'factoryreset' command / apply software

The Generation column lists all of the keys/CSPs and their entry/generation methods. Electronically entered keys are entered by the operator electronically (as defined by FIPS) using the console or a management computer. Pre-configured keys are set as part of the software (hardcoded) and are not operator modifiable.

Externally generated keys are generated outside the module and loaded by the operator electronically and are not compliant with SP 800-133 unless they were generated by another FIPS validated module.

Alternating Bypass Feature

The primary cryptographic function of the module is as a firewall and VPN device. The module implements two forms of alternating bypass for VPN traffic: policy based (for IPsec and SSL VPN) and interface based (for IPsec VPN only).

Policy Based VPN

Firewall policies with IPsec or SSL-VPN mean that the firewall is functioning as a VPN start/end point for the specified source/destination addresses and will encrypt/decrypt traffic according to the policy. Firewall policies with an action of accept mean that the firewall is accepting/sending plaintext data for the specified source/destination addresses.

A firewall policy with an action of accept means that the module is operating in a bypass state for that policy. A firewall policy with IPsec or SSL-VPN means that the module is operating in a non-bypass state for that policy.

Interface Based VPN

Interface based VPN is supported for IPsec only. A virtual interface is created and any traffic routed to the virtual interface is encrypted and sent to the VPN peer. Traffic received from the peer is decrypted. Traffic through the virtual interface is controlled using firewall policies. However, unlike policy based VPN, the action is restricted to Accept or Deny and all traffic controlled by the policy is encrypted/decrypted.

When traffic is routed over the non-virtual interface, the module is operating in a bypass state. When traffic is routed over the virtual interface, the module is operating in a non-bypass state.

In both cases, two independent internal actions must be taken to create a bypass firewall policies.

Key Archiving

The module supports key archiving to a management computer as part of the module configuration file backup. Operator entered keys are archived as part of the module configuration file. The configuration file is stored in plain text, but keys in the configuration file are either AES encrypted using the Configuration Encryption Key or stored as a keyed hash using HMAC SHA-256 using the Configuration Backup Key.

Mitigation of Other Attacks

The module includes a real-time Intrusion Prevention System (IPS) as well as antivirus protection, web content filtering, DNS filtering, application control and data leak prevention. Use of these capabilities is optional.

The FortiOS IPS has two components: a signature based component for detecting attacks passing through the FortiGate appliance and a local attack detection component that protects the firewall from direct attacks. Functionally, signatures are similar to virus definitions, with each signature designed to detect a particular type of attack. The IPS signatures are updated through the FortiGuard IPS service. The IPS engine can also be updated through the FortiGuard IPS service.

FortiOS antivirus protection removes and optionally quarantines files infected by viruses from web (HTTP), file transfer (FTP), and email (POP3, IMAP, and SMTP) content as it passes through the FortiGate modules. FortiOS antivirus protection also controls the blocking of oversized files and supports blocking by file extension. Virus signatures are updated through the FortiGuard antivirus service. The antivirus engine can also be updated through the FortiGuard antivirus service.

FortiOS antispam protection tags (SMTP, IMAP, POP3) or discards (SMTP only) email messages determined to be spam. Multiple spam detection methods are supported including the FortiGuard managed antispam service.

FortiOS web filtering can be configured to provide web (HTTP) content filtering. FortiOS web filtering uses methods such as banned words, address block/exempt lists, and the FortiGuard managed content service.

FortiOS DNS filtering can be configured to provide web content (HTTP/HTTPS) content filtering based on DNS domain lookup. FortiOS DNS filtering uses the FortiGuard DNS database.

FortiOS application control can detect and take action against network traffic depending on the application generating the traffic. FortiOS application control uses the FortiGuard application control database.

FortiOS data leak prevention is used to prevent sensitive data from leaving your network. After sensitive data patterns are defined, data matching the patterns will either be blocked or logged and then allowed.

Whenever a IPS, antivirus, or other filtering event occurs, the modules can record the event in the log and/or send an alert email to an operator.

For complete information refer to the FortiGate Installation Guide for the specific module in question, the FortiGate Administration Guide and the FortiGate IPS Guide.

FIPS 140-2 Compliant Operation

The FortiGate-VM virtual appliance software is shipped in a non-FIPS 140-2 compliant configuration. The following steps must be performed to put the module into a FIPS compliant configuration:

1. Download the model specific FIPS validated software image and md5sum.txt file from the Fortinet Support site at https://support.fortinet.com/
2. Use a hashing utility on the downloaded software image to compare and verify the output against the result from the md5sum.txt file.
3. Install the FIPS validated software image on the hypervisor.
4. Install the entropy token, ensuring that the USB pass-through is enabled in VMware ESXi 6.7. Please be advised that the entropy token **shall** be installed prior to enabling FIPS-CC mode in the next step. Failure to ensure that the entropy token is installed with USB pass-through enabled will result in a non-FIPS compliant module.
5. Enable the FIPS-CC mode of operation as per the "Enabling FIPS-CC Mode" section.

Additional information can be found in the FortiOS 6.4 and 7.0 "FIPS 140-2 and Common Criteria Technote" that can be found on the Fortinet technical documentation website at https://docs.fortinet.com.

In addition, FIPS 140-2 compliant operation requires both that you use the module in its FIPS-CC mode of operation and that you follow secure procedures for installation and operation of the FortiGate-VM software. You must ensure that:

- The FortiGate-VM software is configured in the FIPS-CC mode of operation.
- The FortiGate-VM server is installed in a secure physical location.
- Physical access to the FortiGate-VM server is restricted to authorized operators.
- The Fortinet entropy token is enabled.
- The Fortinet entropy token remains in the server USB port during operation.
- Administrative passwords are at least 8 characters long.
- Administrative passwords are changed regularly.
- Administrator account passwords must have the following characteristics:
  - One (or more) characters must be capitalized
  - One (or more) characters must be lower case
  - One (or more) characters must be numeric
  - One (or more) characters must be non alpha-numeric (e.g. punctuation mark)
- Administration of the module is permitted using only validated administrative methods. These are:
  - Console connection
  - Web-based manager via HTTPS
  - Command line interface (CLI) access via SSH
- Diffie-Hellman groups of less than 2048 bits are not used.
- Client side RSA certificates must use 2048 bit or greater key sizes.
- Only approved and allowed algorithms are used.
- IPSec VPN tunnels using AES-GCM should be configured with a key lifetime of 98,000 KB to ensure a rekey after a maximum of 2^16 encryptions.

The module can be used in either of its two operation modes: NAT/Route or Transparent. NAT/Route mode applies security features between two or more different networks (for example, between a private network and the Internet). Transparent mode applies security features at any point in a network. The current operation mode is displayed on the web-based manager status page and in the output of the `get system status` CLI command.

Once the FIPS validated software has been installed and the module properly configured in the FIPS-CC mode of operation, the module is running in a FIPS compliant configuration. It is the responsibility of the CO to ensure the module only uses approved algorithms and services to maintain the module in a FIPS Approved mode of operation. Using any of the non-approved algorithms and services switches the module to a non-FIPS mode of operation.

Enabling FIPS-CC mode

To enable the FIPS 140-2 compliant mode of operation, the operator must execute the following command from the Local Console:

```
config system fips-cc
   set status enable
end
```

The Operator is required to supply a password for the admin account which will be assigned to the Crypto Officer role.

The supplied password must be at least 8 characters long and correctly verified before the system will restart in FIPS-CC mode.

Upon restart, the module will execute self-tests to ensure the correct initialization of the module's cryptographic functions.

After restarting, the Crypto Officer can confirm that the module is running in FIPS-CC mode by executing the following command from the CLI:

```
get system status
```

If the module is running in FIPS-CC mode, the system status output will display the line:

```
FIPS-CC mode: enable
```

Self-Tests

Startup and Initialization Self-tests

The module executes the following self-tests during startup and initialization:

- Software integrity test using RSA 2048-bit signatures
- Configuration/VPN bypass test using HMAC SHA-256
- AES (128, 256 bit), CBC mode, encrypt known answer test
- AES (128, 256 bit) CBC mode, decrypt known answer test
- AES (128, 256 bit), GCM mode, encrypt known answer test
- AES (128, 256 bit), GCM mode, decrypt known answer test
- HMAC SHA-1 known answer test
- SHA-1 known answer test (tested as part of HMAC SHA-1 known answer test)
- SHA-224 known answer test (tested as part of HMAC-SHA-256 known answer test)
- HMAC SHA-256 known answer test
- SHA-256 known answer test (tested as part of HMAC SHA-256 known answer test)
- HMAC SHA-384 known answer test
- SHA-384 known answer test (tested as part of HMAC SHA-384 known answer test)
- HMAC SHA-512 known answer test
- SHA-512 known answer test (tested as part of HMAC SHA-512 known answer test)
- RSA 2048-bit signature generation known answer test
- RSA 2048-bit signature verification known answer test
- ECDSA pairwise consistency test using P-256 curve
- DRBG known answer tests (as per SP 800-90A)
- Primitive-Z known answer test (KAS-FFC-SSC and KAS-ECC-SSC)
- IKEv1 KDF known answer test
- IKEv2 KDF known answer test
- TLS 1.1 KDF known answer test
- TLS 1.2 KDF known answer test
- SSH KDF known answer test

The results of the startup self-tests are displayed on the console during the startup process.

The startup self-tests can also be initiated on demand using the CLI command **`execute fips kat all`**(to initiate all self-tests) or **`execute fips kat <test>`** (to initiate a specific self-test).

When the self-tests are run, each implementation of an algorithm is tested - i.e. when the AES self-test is run, all AES implementations are tested.

Conditional Self-tests

The module executes the following conditional tests when the related service is invoked:

- Continuous entropy input test
- Continuous DRBG test
- RSA pairwise consistency test
- ECDSA pairwise consistency test
- Configuration/VPN bypass test using HMAC SHA-256
- Software load test using RSA 2048-bit signatures

Critical Function Self-tests

The module also performs the following critical function self-tests applicable to the DRBG, as per NIST SP 800-90A Section 11:

- Instantiate test
- Generate test
- Reseed test

Error State

If any of the self-tests or conditional tests fail, the module enters an error state as shown by the console output below:

```
Self-tests failed
Entering error mode...
The system is going down NOW !!
The system is halted.
```

All data output and cryptographic services are inhibited in the error state.