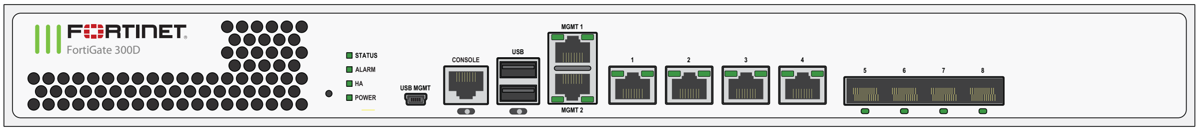
FIPS 140-2 Security Policy

FortiOS™ 5.2



FortiOS™ 5.2 Non-Proprietary FIPS 140-2 Security Policy	
Document Version:	1.7
Publication Date:	October 3, 2016
Description:	Documents FIPS 140-2 Level 1 Security Policy issues, compliancy and requirements for FIPS compliant operation.
Firmware Version:	v5.2.7,build0718,160328

FortiOS™ 5.2 FIPS 140-2 Security Policy

01-525-296259-20151016

Contents

Overview

This document is a FIPS 140-2 Security Policy for Fortinet Incorporated's FortiOS 5.2 firmware, which runs on the FortiGate family of security appliances. This policy describes how the FortiOS 5.2 firmware (hereafter referred to as the 'module') meets 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 module 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 http://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 http://www.fortinet.com/products.
- Find on-line product support for registered products in the technical support section of the Fortinet corporate website at http://www.fortinet.com/support.
- Find contact information for technical or sales related questions in the contacts section of the Fortinet corporate website at http://www.fortinet.com/contact.
- Find security information and bulletins in the FortiGuard Center of the Fortinet corporate website at http://fortiguard.com.

Introduction

The FortiGate product family spans the full range of network environments, from SOHO to service provider, offering cost effective systems for any size of application. FortiGate 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 appliances deliver a full range of network-level services — VPN, intrusion prevention, web filtering, antivirus, antispam and traffic shaping — in dedicated, easily managed platforms.

All FortiGate appliances employ Fortinet's unique FortiASIC content processing chip and the powerful, secure, FortiOS firmware achieve breakthrough price/performance. The unique, ASIC-based architecture analyzes content and behavior in real time, enabling key applications to be deployed right at the network edge where they are most effective at protecting enterprise networks. They can be easily configured to provide antivirus protection, antispam protection and content filtering in conjunction with existing firewall, VPN, and related devices, or as complete network protection systems. The modules support High Availability (HA) in both Active-Active (AA) and Active-Passive (AP) configurations.

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

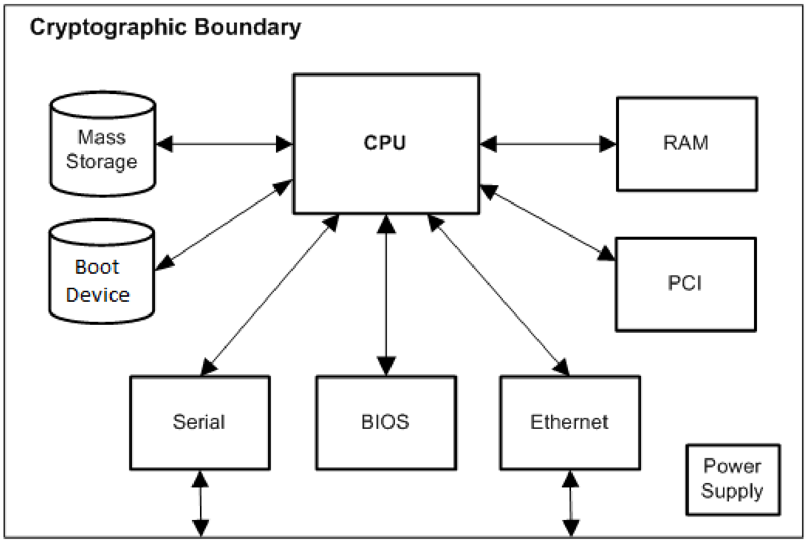
Security Requirement	**Compliance Level**
Cryptographic module Specification	1
Cryptographic module Ports and Interfaces	3
Roles, Services and Authentication	3
Finite State Model	1
Physical Security	1
Operational Environment	N/A
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	3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

Module Description

The module is a firmware operating system that runs exclusively on Fortinet's FortiGate product family. FortiGate units are PC-based, purpose built appliances.

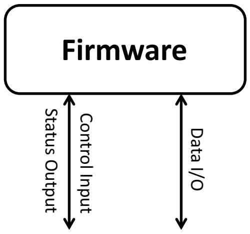
The FortiGate units are multiple chip, standalone cryptographic modules consisting of production grade components contained in a physically protected enclosure.

Figure 1: FortiOS Physical Cryptographic Boundary



The Boot Device in Figure 1 can refer to a separate, internal component or a partition on the Mass Storage device. All references herein of ‘boot device’ shall refer to the configuration specific to the FortiGate appliance.

Figure 2: FortiOS Logical Cryptographic Boundary



For the purposes of FIPS 140-2 conformance testing, the module was tested on a FortiGate-300D unit and used a Fortinet entropy token (FTR-ENT-1) as the entropy source.

The validated firmware version is FortiOS v5.2.7,build0718,160328.

The module can also be executed on any of the following FortiGate/FortiWiFi units and remain vendor affirmed FIPS-compliant

:

FortiGate-20C
FortiGate-30D
FortiGate-40C
FortiGate-60C
FortiGate-60D
FortiGate-60D-PoE
FortiGate-70D
FortiGate-70D-PoE
FortiGate-80C
FortiGate-80CM
FortiGate-80D
FortiGate-90D
FortiGate-90D-PoE
FortiGate-92D
FortiGate-94D-PoE
FortiGate-98D-PoE
FortiGate-100D
FortiGate-110C
FortiGate-111C
FortiGate-140D
FortiGate-140D-PoE
FortiGate-200B
FortiGate-200B-PoE
FortiGate-200D
FortiGate-200D-PoE
FortiGate-240D
FortiGate-240D-PoE
FortiGate-280D-PoE
FortiGate-300C
FortiGate-310B
FortiGate-311B
FortiGate-400D
FortiGate-500D
FortiGate-600C
FortiGate-600D
FortiGate-800C
FortiGate-800D
FortiGate-900D
FortiGate-1000D
FortiGate-1200D
FortiGate-1240B
FortiGate-1500D
FortiGate-3000D
FortiGate-3100D
FortiGate-3016B
FortiGate-3040B
FortiGate-3100D
FortiGate-3140B
FortiGate-3200D
FortiGate-3240C
FortiGate-3600C
FortiGate-3700D
FortiGate-3700DX
FortiGate-3810A
FortiGate-3810D
FortiGate-3815D
FortiGate-3950B
FortiGate-3951B
FortiGate-5001A
FortiGate-5001B
FortiGate-5001BX
FortiGate-5001C
FortiGate-5001D
FortiGate-5101C
FortiSwitch-5203B
FortiWifi-20C
FortiWifi-30D
FortiWifi-30D-PoE
FortiWifi-40C
FortiWifi-60C
FortiWifi-60D
FortiWifi-80CM
FortiWiFi-81CM
FortiWiFi-90D
FortiWiFi-90D-PoE
FortiWiFi-92D
FortiGate-60D-Rugged
FortiGate-100C-Rugged

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

Module Interfaces

The module's logical interfaces and physical ports are described in Table 2.

Table 2: FortiOS logical interfaces and physical ports

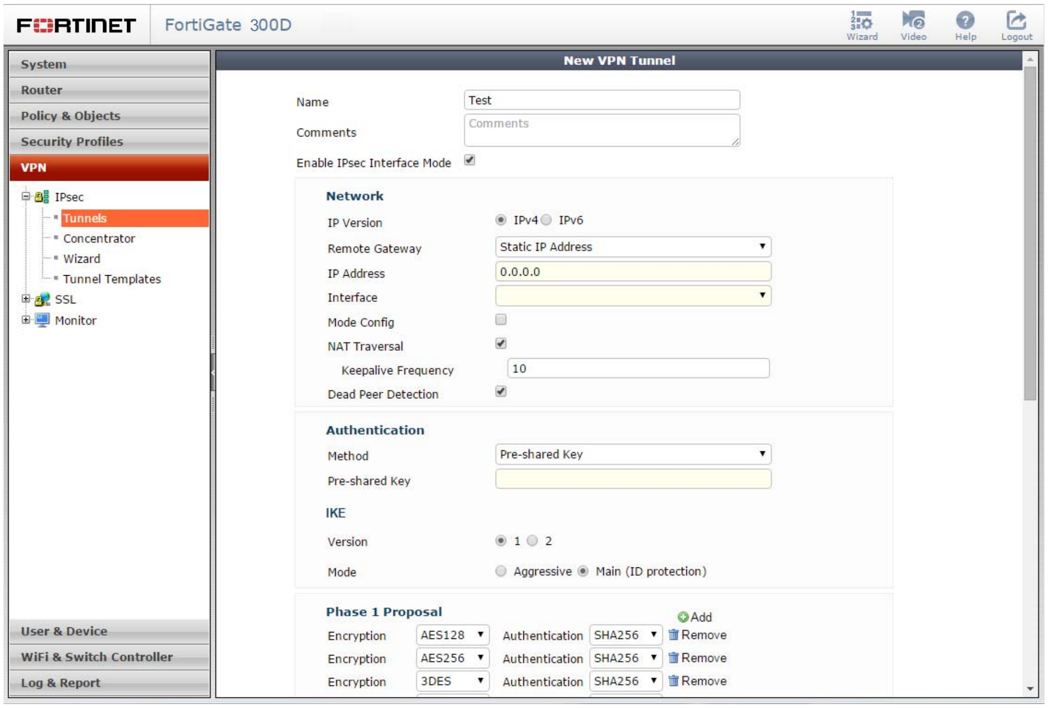
FIPS 140 Interface	Logical Interface	Physical Port
Data Input	API input parameters	Network interface, USB interface (Entropy token)
Data Output	API output parameters	Network interface
Control Input	API function calls	Network interface, serial interface, USB interface (Entropy token)
Status Output	API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

Web-Based Manager

The FortiGate web-based manager provides GUI based access to the module and is the primary tool for configuring the module. 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 (TLS) 1.2 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.

Figure 3: The FortiGate web-based manager



New VPN Tunnel

Name: Test
Comments: Comments
Enable IPsec Interface Mode ☑

Network
IP Version ● IPv4 ○ IPv6
Remote Gateway: Static IP Address
IP Address: 0.0.0.0
Interface:
Mode Config □
NAT Traversal ☑
Keepalive Frequency: 10
Dead Peer Detection ☑

Authentication
Method: Pre-shared Key
Pre-shared Key:

IKE
Version ● 1 ○ 2
Mode ○ Aggressive ● Main (ID protection)

Phase 1 Proposal   Add
Encryption AES128   Authentication SHA256   Remove
Encryption AES256   Authentication SHA256   Remove
Encryption 3DES   Authentication SHA256   Remove

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 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 or read only access permissions including the ability to create operator accounts.

The module also provides a **Network User** role for end-users (Users). Network Users can make use of the encrypt/decrypt services, but cannot access the module 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
authenticate to module	WE	Operator Password, Diffie-Hellman Key, HTTP/TLS and SSH Server/Host Keys, HTTPS/TLS and SSH Session Authentication Keys, and HTTPS/TLS Session Encryption Keys, DRBG Output, DRBG Seed, DRBG Input String
show system status	WE	N/A
show FIPS-CC mode enabled/disabled (console/CLI only)	WE	N/A
enable FIPS-CC mode of operation (console only)	WE	Configuration Integrity Key
key zeroization	WE	All Keys, See “Key Zeroization” on page 11
execute factory reset (disable FIPS-CC mode, console/CLI only)	E	All keys except firmware update key, configuration integrity key, configuration backup key
execute FIPS-CC on-demand self-tests (console only)	E	Configuration Integrity Key, Firmware Integrity Key
add/delete operators and network users	WE	Crypto Officer Password, Network User Password
set/reset operator and network user passwords	WE	Crypto Officer Password, Network User Password
backup/restore configuration file	WE	Configuration Encryption Key, Configuration Backup Key
read/set/delete/modify module configuration	WE	N/A
execute firmware update	E	Firmware Update Key
read log data	WE	N/A
delete log data (console/CLI only)	WE	N/A
execute system diagnostics (console/CLI only)	WE	N/A

Table 3: Services available to Crypto Officers

Service	Access	Key/CSP
enable/disable alternating bypass mode	WE	N/A
read/set/delete/modify IPSec/SSL VPN configuration	N/A	IPSec: IPSec Manual Authentication Key, IPSec Manual Encryption Key, IKE Pre-Shared Key, IKE RSA Key SSL: HTTPS/TLS Server/Host Key, HTTPS/TLS Session Authentication Key, HTTPS/TLS SSH Session Encryption Key
read/set/modify HA configuration	WE	HA Password, HA Encryption Key

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

Service/CSP	Access	Key/CSP
authenticate to module	WE	Network User Password, Diffie-Hellman Key, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Authentication Key, HTTPS/TLS Session Encryption Key, DRBG Output, DRBG Seed, DRBG Input String
IPSec VPN controlled by firewall policies	E	Diffie-Hellman Key, IKE and IPSec Keys, DRBG Output, DRBG Seed, DRBG Input String
SSL VPN controlled by firewall policies	E	Network User Password, Diffie-Hellman Key, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Authentication Key, HTTPS/TLS Session Encryption Key, DRBG Output, DRBG Seed, DRBG Input String

Non-FIPS Approved Services

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

- Configuration backups using password protection
- LLTP and PPTP VPN

All services in Table 3 and Table 4 are considered non-approved when using the following algorithms:

- Non-compliant-strength Diffie-Hellman
- Non-compliant-strength RSA key wrapping
- DES
- HMAC-MD5

Authentication

The module implements identity based authentication. Operators must authenticate with a user-id and password combination to access the modules remotely or locally via the console. Remote operator 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.

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.

Note that operator 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 94^8 which is less than 1/100,000.

Operator 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/(948/108,000) which is less than 1/100,000

The minimum password length is 8 characters when in FIPS-CC mode (maximum password length is 32 characters). The password may contain any combination of upper- and lower-case letters, numbers, and printable symbols; allowing for 94 possible characters. The odds of guessing a password are 1 in 94^8 which is significantly lower than one in a million. Recommended procedures to increase the password strength are explained in "FIPS 140-2 Compliant Operation" on page 17.

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 or IKE RSA key (RSA 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 2048bit RSA key size)

Therefore the minimum odds of guessing the authentication key for IPSec is 1 in 94^8, based on the IKE Pre-shared key.

Physical Security

The physical security for the module is provided by the FortiGate hardware which uses production grade components and an opaque enclosure.

Operational Environment

The module constitutes the entire firmware operating system for a FortiGate unit and can only be installed and run on a FortiGate unit. The module provides a proprietary and non-modifiable operating system and does not provide a programming environment.

For the purposes of FIPS 140-2 conformance testing, the module was tested on a FortiGate-300D unit.

Cryptographic Key Management

Random Number Generation

The module uses a firmware 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. There is no assurance of the minimum strength of generated keys.

Entropy Token

The module uses a Fortinet entropy token (part number FTR-ENT-1 or part number FTR-ENT-2) to seed the DRBG during the module's 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.

The default reseed period is once every 24 hours (1440 minutes). The entropy token must be installed to complete the boot process and to reseed the DRBG. The entropy token is responsible for loading a minimum of 256 bits of entropy.

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 erasing the module's boot device and then power cycling the FortiGate unit. To erase the boot device, execute the following command from the CLI:

```
execute erase-disk <boot device>
```

The boot device ID may vary depending on the FortiGate module. Executing the following command will output a list of the available internal disks:

```
execute erase-disk ?
```

Algorithms

Table 5: FIPS Approved Algorithms

Algorithm	**NIST Certificate Number**
CTR DRBG (NIST SP 800-90A) with 256-bits	1161
Triple-DES in CBC mode with 192-bits	2172, 2173
AES in CBC mode (128-, 192-, 256-bits)	3963, 3964
SHA-1	3267, 3268
SHA-256	3267, 3268
SHA-384	3267, 3268
SHA-512	3268, 3268
HMAC SHA-1	2581, 2582
HMAC SHA-256	2581, 2582
HMAC SHA-384	2581, 2582
HMAC SHA-512	2581, 2582
RSA PKCS1 -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	2024
CVL (SSH) - with TDES-192 bit-CBC, AES 128 bit-, AES 256 bit -CBC (using SHA1)	794
CVL (TLS) - TLS 1.1 and 1.2	794
CVL (IKE v1 and v2)	795

Table 6: FIPS Allowed Algorithms

Algorithm
RSA (key wrapping; key establishment methodology provides 112 or 128 bits of encryption strength)
Diffie-Hellman (key agreement; key establishment methodology provides between 112 and 201 bits of encryption strength)
NDRNG (Entropy Token) - please refer to the "Entropy Token" on page 11 for additional information.

Table 7: FIPS Non-Compliant Algorithms

Algorithm
SHA-384
SHA-512
HMAC SHA-384
HMAC SHA-512

Note that the algorithms in Table 7 are listed as non-compliant since Known Answer Tests for the algorithms are not implemented in the validated firmware build.

Table 8: Non-FIPS Approved Algorithms

Algorithm
DES (disabled in FIPS-CC mode)
MD5 (disabled in FIPS-CC mode)
HMAC-MD5 (disabled in FIPS-CC mode)
RSA 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.
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.

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

Cryptographic Keys and Critical Security Parameters

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

Key or CSP	The key or CSP description.
Storage	Where and how the keys are stored
Usage	How the keys are used
Zeroization	The key zeroization method

Table 9: Cryptographic Keys and Critical Security Parameters used in FIPS-CC mode

Key or CSP	Generation	Storage	Usage	Zeroization
NDRNG output string	Automatic	Boot device Plain-text	Input string for the entropy pool	By erasing the Boot device and power cycling the module
DRBG seed	Automatic	Boot device Plain-text	Seed used by the DRBG (output from NDRNG)	By erasing the Boot device and power cycling the module
DRBG output	Automatic	Boot device Plain-text	Random numbers used in cryptographic algorithms	By erasing the Boot device and power cycling the module
DRBG v and key values	Automatic	Boot device Plain-text	Internal state values for the DRBG	By erasing the Boot device and power cycling the module
IPSec Manual Authentication Key	Manual	Boot device AES encrypted	Used as IPSec Session Authentication Key	By erasing the Boot device and power cycling the module
IPSec Manual Encryption Key	Automatic	SDRAM Plain-text	Used as IPSec Session Encryption Key	By erasing the Boot device and power cycling the module
IPSec Session Authentication Key	Automatic	SDRAM Plain-text	IPSec peer-to-peer authentication using HMAC SHA-1 or HMAC SHA-256	By erasing the Boot device and power cycling the module
IPSec Session Encryption Key	Automatic	SDRAM Plain-text	VPN traffic encryption/decryption using Triple-DES or AES	By erasing the Boot device and power cycling the module

Table 9: Cryptographic Keys and Critical Security Parameters used in FIPS-CC mode

Key or CSP	Generation	Storage	Usage	Zeroization
IKE Pre-Shared Key	Manual	Boot device AES encrypted	Used to generate IKE protocol keys	By erasing the Boot device and power cycling the module
IKE Authentication Key	Automatic	SDRAM Plain-text	IKE peer-to-peer authentication using HMAC SHA-1 , -256, -384 or -512 (SKEYID_A)	By erasing the boot device and power cycling the module
IKE Key Generation Key	Automatic	SDRAM Plain-text	IPSec SA keying material (SKEYID_D)	By erasing the boot device and power cycling the module
IKE Session Encryption Key	Automatic	SDRAM Plain-text	Encryption of IKE peer-to-peer key negotiation using Triple-DES or AES (SKEYID_E)	By erasing the boot device and power cycling the module
IKE RSA Key	Manual	Boot device Plain text	Used to generate IKE protocol keys	By erasing the boot device and power cycling the module
Diffie-Hellman Keys	Automatic	SDRAM Plaintext	Key agreement and key establishment	By erasing the boot device and power cycling the module
Firmware Update Key	Preconfigured	Boot device Plain-text	Verification of firmware integrity when updating to new firmware versions using RSA public key (firmware load test)	By erasing the boot device and power cycling the module
Firmware Integrity Key	Preconfigured	Boot device Plain-text	Verification of firmware integrity in the firmware integrity test using RSA public key (firmware integrity test)	By erasing the boot device and power cycling the module
HTTPS/TLS Server/Host Key	Preconfigured	Boot device Plain-text	RSA private key used in the HTTPS/TLS protocols (key establishment)	By erasing the boot device and power cycling the module
HTTPS/TLS Session Authentication Key	Automatic	SDRAM Plain-text	HMAC SHA-1, -256 or -384 key used for HTTPS/TLS session authentication	By erasing the boot device and power cycling the module
HTTPS/TLS Session Encryption Key	Automatic	SDRAM Plain-text	AES or Triple-DES key used for HTTPS/TLS session encryption	By erasing the boot device and power cycling the module
SSH Server/Host Key	Preconfigured	Boot device Plain-text	RSA private key used in the SSH protocol (key establishment)	By erasing the boot device and power cycling the module
SSH Session Authentication Key	Automatic	SDRAM Plain-text	HMAC SHA-1 or HMAC SHA-256 key used for SSH session authentication	By erasing the boot device and power cycling the module

Table 9: Cryptographic Keys and Critical Security Parameters used in FIPS-CC mode

Key or CSP	Generation	Storage	Usage	Zeroization
SSH Session Encryption Key	Automatic	SDRAM Plain-text	AES or Triple-DES key used for SSH session encryption	By erasing the boot device and power cycling the module
Crypto Officer Password	Manual	Boot device SHA-1 hash	Used to authenticate operator access to the module	By erasing the boot device and power cycling the module
Configuration Integrity Key	Preconfigured	Boot device Plain-text	HMAC SHA-256 hash used for configuration integrity test	By erasing the boot device and power cycling the module
Configuration Encryption Key	Automatic	Boot device Plain-text	AES 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 erasing the boot device and power cycling the module
Configuration Backup Key	Automatic	Boot device Plain-text	HMAC SHA-256 key used to encrypt crypto officer passwords in the backup configuration file	By erasing the boot device and power cycling the unit
Network User Password	Manual	Boot device SHA-1 hash	Used to authenticate network access to the module	By erasing the boot device and power cycling the unit
HA Password	Manual	Boot device AES encrypted	Used to authenticate FortiGate units in an HA cluster	By erasing the boot device and power cycling the unit
HA Encryption Key	Manual	Boot device AES encrypted	Encryption of traffic between units in an HA cluster using AES	By erasing the boot device and power cycling the unit

Note: The Generation column lists all of the keys/CSPs and their entry/generation methods. Manual 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 firmware (hardcoded) and are not operator modifiable. Automatic keys are generated as part of the associated protocol.

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 route/interface based (for IPSec VPN only).

Policy Based IPSec VPN

Firewall policies with an action of IPSec mean that the firewall is functioning as an IPSec VPN start/end point for the specified source/destination addresses and is operating in a non-bypass state.

Firewall policies with an action of accept and use the ssl.root interface mean the firewall is acting as an SSL VPN end point for SSL VPN clients and is operating in a non-bypass state.

Firewall policies with an action of accept mean that the firewall is accepting/sending plaintext data for the specified source/destination addresses and is operating in a bypass state.

Route/Interface Based IPSec VPN

Route/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 (policy based and route/interface based VPN), two independent actions must be taken by a CO to create bypass firewall policies: the CO must create the bypass policy and then specifically enable that policy.

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, antispam and content filtering. 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.

Whenever a IPS, antivirus, antispam or 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

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 unit. You must ensure that:

- The FortiGate unit is configured in the FIPS-CC mode of operation.
- The FortiGate unit is installed in a secure physical location.
- Physical access to the FortiGate unit is restricted to authorized operators.
- The Fortinet entropy token is used to seed the DRBG.
- The Fortinet entropy token remains in the USB port during operation
- Administrative and Network User passwords are at least 8 characters long.
- Administrative and Network User passwords are changed regularly.
- Administrative and Network User passwords must have the following characteristics:
  - One (or more) of the characters must be capitalized
  - One (or more) of the characters must be numeric
  - One (or more) of the 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 (see “Algorithms” on page 12).

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.

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 entropy-token enable
  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

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

- Firmware integrity test using RSA signatures
- Configuration/VPN bypass test using HMAC SHA-256
- Triple-DES, CBC mode, encrypt known answer test
- Triple-DES, CBC mode, decrypt known answer test
- AES, CBC mode, encrypt known answer test
- AES, CBC 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)
- HMAC SHA-256 known answer test
- SHA-256 known answer test (tested as part of HMAC SHA-256 known answer test)
- RSA signature generation known answer test
- RSA signature verification known answer test
- DRBG 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 - e.g. when the AES self-test is run, all AES implementations are tested.

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

- Continuous NDRNG test
- Continuous DRBG test
- RSA pairwise consistency test
- Configuration/VPN bypass test using HMAC SHA-256
- Firmware load test using RSA signatures

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.