Fortinet Inc.

FortiClient Crypto Module v1.0

FIPS 140-3 Non-Proprietary Security Policy

Version: 1.0

Prepared for:

FORTINET

Prepared by:

TERON LABS

www.teronlabs.com

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is a FIPS 140-3 Security Policy for Fortinet's FortiClient Crypto Module version 1.0 for Windows running inside FortiClient 7.4 and 8.0. This policy describes how the FortiClient Crypto Module (hereafter referred to as the 'module') meets the FIPS 140-3 security requirements and how to operate the module in a FIPS compliant manner. This policy was created as part of the FIPS 140-3 Level 1 validation of the module.

The Federal Information Processing Standards Publication 140-3 - Security Requirements for Cryptographic Equipment (FIPS 140-3) details the United States Federal Government requirements for cryptographic equipment. Detailed information about the FIPS 140-3 standard and validation program is available on the NIST (National Institute of Standards and Technology) website at https://csrc.nist.gov/projects/cryptographic-module-validation-program

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	1
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	N/A
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

1.3 Additional Information

This policy deals specifically with operation and implementation of the module in the technical terms of the FIPS 140-3 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://support.fortinet.com/.
- 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://wwww.fortiguard.com.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The FortiClient Crypto Module v1.0 (hereafter referred to as the module) is a software cryptographic module that provides cryptographic services for FortiClient, Fortinet's endpoint security application. FortiClient and the module are designed to execute on a general-purpose computer (GPC) hardware platform. The module has no physical characteristics and relies on the physical characteristics of the GPC on which it runs.

The module is a component of the FortiClient software and requires the following:

- A commercially available, general purpose, x86 compatible computer
- Windows 11

The purpose of the module is to provide cryptographic services for FortiClient.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The module's cryptographic boundary encompasses the following FortiClient software binaries:

- FCCryptd.exe
- FCCrypt.dll, which serves as a wrapper for:
    - libcrypto-3-x64.dll/libcrypto-3-arm64.dll
    - libssl-3-x64.dll/libssl-3-arm64.dll

Tested Operational Environment's Physical Perimeter (TOEPP):

The TOEPP refers to the physical perimeter of the chassis of the general-purpose computer (GPC) on which the module is running.

The entire box in Figure 1 represents the TOEPP, while the gray boxes within the TOEPP represent the cryptographic boundary. For details on the cryptographic boundary and the Tested Operational Environment's Physical Perimeter, please refer to sections 2.1.6 and 2.1.7, respectively

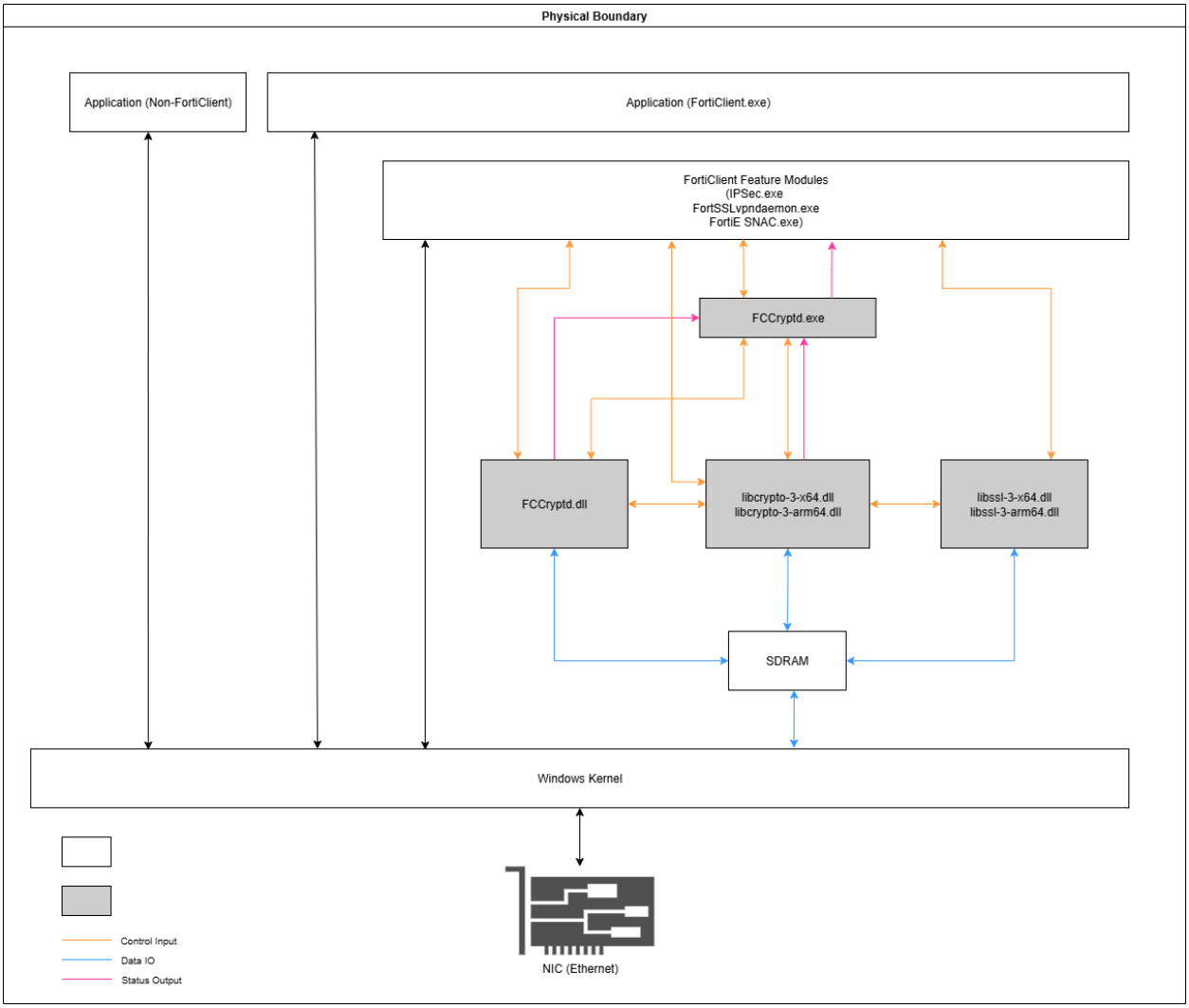

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

N/A for this module.

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

Package or File Name	Software/ Firmware Version	Features	Integrity Test
Fccryptd.exe Fccryptd.dll	7.4.4		RSA-4096

Package or File Name	Software/ Firmware Version	Features	Integrity Test
libcrypto-3-x64.dll libssl-3-x64.dll			

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Microsoft Windows 11 (64-bit)	Dell Inc. Latitude 5440	13th Gen Intel(R) Core(TM) i5-1345U	No		7.4.4

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform
Microsoft Windows 11	GPC with x86 or x86-64 bit processor and FortiClient 7.4 or 8.0

Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

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

2.3 Excluded Components

None.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
FIPS Deployed Package	The module is deployed as a FIPS package on the client PC	Approved	Run the command: fccryptd.exe fips version

Table 5: Modes List and Description

The module does not support multiple modes of operation. The validated configuration of the module meets the FIPS 140-3 requirements by default once the start-up procedures described in section 11.1 have been performed - i.e. the module has no FIPS "mode" that can be enabled or disabled by the user or local administrator and is always in the approved mode of operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A7626	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A7626	Direction - Decrypt, Encrypt Key Length - 128	SP 800-38A
AES-GCM	A7626	Direction - Decrypt, Encrypt IV Generation - External IV Generation Mode - 8.2.1 Key Length - 128, 256 Tag Length - 128 IV Length - IV Length: 96 Payload Length - Payload Length: 16, 128, 136, 256, 264 AAD Length - AAD Length: 0, 128, 136, 256	SP 800-38D
Counter DRBG	A7626	Prediction Resistance - No Supports Reseed - Yes Mode - AES-256 Derivation Function Enabled - Yes Additional Input - Additional Input: 0-256 Increment 256 Entropy Input - Entropy Input: 256 Nonce - Nonce: 128 Personalization String Length - Personalization String Length: 0-256 Increment 256 Returned Bits - 256	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A7626	Curve - P-256, P-384, P-521 Secret Generation Mode - extra bits, testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A7626	Curve - P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A7626	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigVer (FIPS186-5)	A7626	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
HMAC-SHA-1	A7626	MAC - MAC: 32-160 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
HMAC-SHA2-256	A7626	MAC - MAC: 32-256 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
HMAC-SHA2-384	A7626	MAC - MAC: 32-384 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
HMAC-SHA2-512	A7626	MAC - MAC: 32-512 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
HMAC-SHA3-256	A7626	MAC - MAC: 32-256 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
HMAC-SHA3-384	A7626	MAC - MAC: 32-384 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
HMAC-SHA3-512	A7626	MAC - MAC: 32-512 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A7626	Domain Parameter Generation Methods - P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A7626	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
PBKDF	A7626	Iteration Count - Iteration Count: 1-1000 Increment 1 HMAC Algorithm - SHA-1 Password Length - Password Length: 8-31 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 112-512 Increment 8	SP 800-132
RSA KeyGen (FIPS186-5)	A7626	Key Generation Mode - probable Modulo - 2048, 3072, 4096 p mod 8 - 0 Primality Tests - 2powSecStr q mod 8 - 0	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
		Info Generated By Server - No Private Key Format - standard Public Exponent Mode - random	
RSA SigGen (FIPS186-5)	A7626	Hash Pair - Hash Algorithm - SHA2-256 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A7626	Hash Pair - Hash Algorithm - SHA2-256 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5 Public Exponent Mode - random	FIPS 186-5
Safe Primes Key Generation	A7626	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3
SHA2-256	A7626	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-384	A7626	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA2-512	A7626	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA3-256	A4973	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-256	A7626	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A7626	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-512	A7626	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202

Table 6: Approved Algorithms

* Note: The recommended iteration count of 1000 is used, as per SP800-132.

** Note: Keys derived from passwords, as described in SP800-132, may only be used in storage applications.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type:Symmetric and Asymmetric	N/A	SP 800-133 Rev.2 Section 4, example 1 direct output from DRBG.

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Not Applicable.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

None.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

None.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
AES for Symmetric Encryption/Decryption	BC-UnAuth	Encryption/Decryption	SP 800-38A:null	AES-CBC: (A7626) Key Length: 128, 192, 256 AES-ECB: (A7626) Key Length: 128 AES-GCM: (A7626)
DRBG	DRBG	Deterministic Random Bit Generation	SP 800-90Arev1:null	Counter DRBG: (A7626) AES-CBC: (A7626) Key Length: 256

Name	Type	Description	Properties	Algorithms
				Prediction Resistance: No Derivation Function: Yes
ECDSA for Key Generation	AsymKeyPair-KeyGen	Key Generation of the ECDSA Private Key and ECDSA Public Key	FIPS 186-5:null	ECDSA KeyGen (FIPS186-5): (A7626) Curves: P-256, P-384, P-521 CKG: ()
ECDSA for Key Verification	AsymKeyPair-KeyVer	ECDSA Public Key Verification	FIPS 186-5:null	ECDSA KeyVer (FIPS186-5): (A7626) Curves: K-233, P-256, P-384, P-521
ECDSA for Signature Generation	DigSig-SigGen	ECDSA Signature Generation for the ECDSA Private Key	FIPS 186-5:null	ECDSA SigGen (FIPS186-5): (A7626) Curves: P-256, P-384, P-521 SHA2-256: (A7626) SHA2-384: (A7626) SHA2-512: (A7626)
ECDSA for Signature Verification	DigSig-SigVer	ECDSA Signature Verification for the ECDSA Public Key	FIPS 186-5:null	ECDSA SigVer (FIPS186-5): (A7626) Curves: P-256, P-384, P-521 SHA2-256: (A7626) SHA2-384: (A7626)

Name	Type	Description	Properties	Algorithms
				SHA2-512: (A7626)
HMAC for Keyed Hash Operation	MAC	Message Authentication	FIPS 198-1:null	HMAC-SHA-1: (A7626) MAC Lengths: 160, 256, 384, 512 HMAC-SHA2-256: (A7626) HMAC-SHA2-384: (A7626) HMAC-SHA2-512: (A7626) HMAC-SHA3-512: (A7626) HMAC-SHA3-256: (A7626) HMAC-SHA3-384: (A7626)
ECC-SSC	KAS-SSC	ECDH Shared Secret Computation	SP 800-56Ar3:null	KAS-ECC-SSC Sp800-56Ar3: (A7626) Curves: P-256, P-384, P-521
FFC-SSC	KAS-SSC	DH Shared Secret Computation	SP 800-56Ar3:null	KAS-FFC-SSC Sp800-56Ar3: (A7626) Safe Primes: MODP-2048, MODP-3072, MODP-4096,

Name	Type	Description	Properties	Algorithms
				MODP-6144, MODP-8192
PBKDF	PBKDF	Password-Based Key Derivation	SP 800-132:null	PBKDF: (A7626) HMAC-SHA-1: (A7626) Key Length: 8-105 Increment 8
RSA for Key Generation	CKG	Key Pair Generation of the RSA Private Key and RSA Public Key	FIPS 186-5:null	RSA KeyGen (FIPS186-5): (A7626) Modulo Sizes: 2048, 3072, 4096 CKG: ()
RSA for Signature Generation	DigSig-SigGen	RSA Signature Generation for the RSA Private Key	FIPS 186-5:null	RSA SigGen (FIPS186-5): (A7626) Modulo Size: 2048, 3072, 4096 SHA2-256: (A7626) SHA2-384: (A7626) SHA2-512: (A7626)
RSA for Signature Verification	DigSig-SigVer	RSA Signature Verification for the RSA Public Key	FIPS 186-5:null	RSA SigVer (FIPS186-5): (A7626) Modulo Sizes: 2048, 3072, 4096 SHA2-256: (A7626) SHA2-384: (A7626) SHA2-512: (A7626)

Name	Type	Description	Properties	Algorithms
SHA for Message Digest	SHA	Message Digest	FIPS 180-4,:null FIPS 202:null	SHA2-256: (A7626) SHA2-384: (A7626) SHA2-512: (A7626) SHA3-512: (A7626) SHA3-256: (A7626) SHA3-384: (A7626)
RSA and SHA for Integrity Test	DigSig-SigVer	RSA and SHA for the pre-operational integrity tests.	FIPS 198-1,:null FIPS 186-5:null	RSA SigVer (FIPS186-5): (A7626) Modulo Size: 4096 SHA2-256: (A7626)
Safe Prime Key Generation	CKG	Key Generation for all module services that utilize KAS-FFC-SSC	SP 800-56Arev3:null	Safe Primes Key Generation: (A7626) Safe Primes: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192
Symmetric Key Generation for AES	CKG			Counter DRBG: (A7626) CKG: ()
KAS Key Pair Generation	AsymKeyPair-KeyGen	Key pair generation for KAS ephemeral key pairs		Safe Primes Key Generation: (A7626) ECDSA KeyGen (FIPS186-

Name	Type	Description	Properties	Algorithms
				5): (A7626) ECDSA KeyVer (FIPS186-5): (A7626) Counter DRBG: (A7626) CKG: ()
Entropy Generation	ENT-ESV	Entropy source		SHA3-256: (A4973)

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

Symmetric Keys:
The module generates symmetric keys as per section 6.2.3 of SP800-133rev2 using PBKDF2 (Password Based Key Derivation Function).

PBKDF:
PBKDF2 uses SHA-1 and a 128-bit salt to derive keys, which are only used for storage purposes in accordance with SP800-132 and IG D.N Option 1a.

The probability of guessing a password at random (S) is equal to the Character Pool Size (C) raised to the power of the number of characters in the password (N). The Character Pool Size (C) is 95, as the password can contain lowercase (a- z) [26], uppercase (A-Z) [26], digits (0-9) [10], and special characters (!"$,) [33]. The minimum password length is 8 characters and is enforced. However, there is no constraint on the number of times a character is used from the character pool. Therefore, the probability of guessing a password (S) is between 95^8 and 95^105 – i.e. using an 8-character password the probability of guessing the password is 1 in 6,634,204,312,890,625.

Asymmetric Keys:
ECDSA and RSA key generation is done using RBG provided as per section 4 of SP800-133rev2, and the elliptic curve is used with appropriate base order to meet SP800-57 Part 1 requirements.

Key agreement:
The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer some of the underlying KAS cryptographic functionality to be used by an external operator/application as part of an approved KAS.

The module supports Diffie Hellman Shared Secret Computation for KAS-ECC-SSC Ephemeral Unified and KAS-FFC-SSC Ephemeral DH key agreement schemes.

The module supports FFC full public key validation per Section 5.6.2.3.1 SP 800-56Arev3 and ECC full public key validation per Section 5.6.2.3.3 SP 800-56Arev3.

2.8 RBG and Entropy

Cert Number	Vendor Name
E138	Fortinet Inc.

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
FortiClient CPU Jitter Entropy Library	Non-Physical	Windows 11 (64-bit) on 13th Gen Intel(R) Core(TM) i5-1345U	256	256	SHA3-256 (A4973)

Table 10: Entropy Sources

The module uses FortiClient CPU Jitter Entropy Library 1.0 to seed the DRBG during the module's boot process and to periodically reseed the DRBG. The entropy loaded into the approved AES-256-bit DRBG is 256 bits. The noise source is over-sampled and then a SHA3-256 conditioning component is applied as part of the entropy source. The default reseed period is once every 24 hours (1440 minutes) and is configurable (1 to 1440 minutes).

The module uses CTR_DRBG as per section 10.2.1 of SP800-90Arev1 to generate random numbers.

2.9 Key Generation

The module complies with sections 4, 5, 6.1, and 6.2.3 of SP800-133rev2, where the module uses its Approved DRBG to generate random values and seeds used for symmetric key generation. The resulting symmetric key or generated seed is an unmodified output from the DRBG.

2.10 Key Establishment

For KAS-ECC-SSC, elliptic-curve groups approved for use by the key-establishment schemes specified in SP800- 56Arev3 Appendix D are used.

For KAS-FFC-SSC, an approved safe prime group listed in SP800-56Arev3 Appendix D is used.

2.11 Industry Protocols

The module does not implement any industry protocols.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

The module is a software module that communicates with the FortiClient software and the Windows Kernel, as such the module does not have any physical interfaces. The module's logical interfaces and the types of data passed over the interfaces are described below.

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API input parameters for data- plain text and cipher text, digital signatures, public keys
N/A	Data Output	API output parameters for data- encrypted or decrypted, digital signatures, hashes
N/A	Control Input	API input commands from FortiClient application, entropy input
N/A	Status Output	API return values to the FortiClient application

Table 11: Ports and Interfaces

Note: Data I/O occurs between the module and the FortiClient application and the Windows Kernel (connection to NIC for VPN tunnel). All the data output via the data output interface is inhibited when the module is performing pre-operational tests, zeroization, or enters an error state. The module does not support a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module does not claim any authentication methods as it is a FIPS 140-3 level 1 software module.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 12: Roles

The module provides the following roles:

- Crypto Officer (CO)

A User with Windows administrative privileges assumes the "Crypto Officer" role. Multiple accounts can be logged in to the Windows PC, but only one account (and therefore only one role) can be active at a time through Windows user switching.

The module does not provide a Maintenance role.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Asymmetric Key Generation	RSA and ECDSA Keys are Generated	Successful completion of service. IG 2.4.C, Scenario 2	key size ECDSA: curve identifier. RSA: domain parameters.	Key pair struct (public + private keys), status	ECDSA for Key Generation ECDSA for Key Verification RSA for Key Generation KAS Key Pair Generation	Crypto Officer - ECDSA Public Key: G - ECDSA Private Key: G - RSA Public Key: G - RSA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Private Key: G - DH Private Key: G - DH Public Key: G - ECDH Private Key: G - ECDH Public Key: G
Derive Key Via PBKDF2	The module derives key using user password	Successful completion of service. IG 2.4.C, Scenario 2	Password, salt, iteration count, key length, provider context	Key is derived, status	HMAC for Keyed Hash Operation PBKDF SHA for Message Digest	Crypto Officer - PBKDF2 Derived Key: G - User Password: W,E
Execute Manual Self-Tests	Manual Self-Tests are executed	Successful completion of service. IG 2.4.C, Scenario 2	Windows CLI Command	Self-test results displayed in Command Prompt Window	AES for Symmetric Encryption/Decryption DRBG ECDSA for Key Generation ECDSA for Key Verification ECDSA for Signature Generation ECDSA for Signature Verification HMAC for Keyed Hash	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Operation ECC-SSC FFC-SSC PBKDF RSA for Key Generation RSA for Signature Generation RSA for Signature Verification SHA for Message Digest RSA and SHA for Integrity Test Safe Prime Key Generation Symmetric Key Generation for AES Entropy Generation	
Generate Message Digest	Message Digests are generated using SHA algorithms	Successful completion of service. IG 2.4.C, Scenario 2	Message	Message digest, status	SHA for Message Digest	Crypto Officer
Generate Keyed Hash	Keyed hash is generated to provide message authentication	Successful completion of service. IG 2.4.C, Scenario 2	key struct, message	MAC value, MAC length, status	HMAC for Keyed Hash Operation	Crypto Officer - HMAC Key: E
Install	FortiClient module installed	Successful complet	Module binary, Windows	FortiClient installed on GPC		Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	on the GPC	ion of service. IG 2.4.C, Scenario 2	GUI installation commands			
Initialize	FortiClient module installed on the GPC	Successful completion of service. IG 2.4.C, Scenario 2	Windows GUI	Module initialized as part of the FortiClient startup		Crypto Officer
Key Agreement	Shared secret computation using FFC DH and ECC DH	Successful completion of service. IG 2.4.C, Scenario 2	Key structs (key agreement keys)	Shared secret, status	ECC-SSC FFC-SSC Safe Prime Key Generation	Crypto Officer - ECDH Shared Secret Key: G - ECDH Private Key: G,E - ECDH Public Key: G,E - DH Private Key: G,E - DH Public Key: G,E - DH Shared Secret Key: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Perform Digital Signature	Digital Signature generation and verification using RSA and ECDSA asymmetric keys	Successful completion of service. IG 2.4.C, Scenario 2	Signing key, message, signature parameters	Signature value, status	ECDSA for Signature Generation ECDSA for Signature Verification RSA for Signature Generation RSA for Signature Verification	Crypto Officer - ECDSA Public Key: E - ECDSA Private Key: E - RSA Public Key: E - RSA Private Key: E
Random Bit Generation	A counter-based random bit is generated using JitterEntropy	Successful completion of service. IG 2.4.C, Scenario 2	DRBG struct (RGB state), entropy input, requested big length, additional input, personalization string	Random value, status, RBG state updated	DRBG	Crypto Officer - Entropy Input String: E - DRBG Seed: E - DRBG 'V' Value: E - DRBG 'Key' Value: E
Show Module Status	Module Status Displayed	Successful completion of service. IG 2.4.C, Scenario 2	FCCryptd.exe fips version	'FIPS' status message, integrity test results, module version		Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show Version Information	Crypto Module version shown	Successful completion of service. IG 2.4.C, Scenario 2	FCCryptd.exe fips version	'FIPS' status message, integrity test results, module version		Crypto Officer
Symmetric Key Encryption/Decryption	AES Keys are used to encrypt plaintext and decrypt ciphertext	Successful completion of service. IG 2.4.C, Scenario 2	Encryption or decryption key, plaintext or ciphertext data, cipher parameters	Ciphertext or plaintext, authentication tag (authenticated modes), status	AES for Symmetric Encryption/Decryption	Crypto Officer - AES Key: E
Symmetric Key Generation	AES Keys are Generated	Successful completion of service. IG 2.4.C, Scenario 2	Key length, provider context	Key struct, Symmetric key generated, status	Symmetric Key Generation for AES	Crypto Officer - AES Key: G
Key Zeroization	All Sensitive Security are zeroized	Successful completion of service. IG 2.4.C, Scenario 2	Key struct(s)	Zeroized key material, status		Crypto Officer - AES Key: Z - DH Private Key: Z - DH Public Key: Z - DH Shared Secret Key: Z - Entrop

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						y Input String: Z - DRBG Seed: Z - DRBG 'V' Value: Z - DRBG 'Key' Value: Z - ECDH Private Key: Z - ECDH Public Key: Z - ECDH Shared Secret Key: Z - ECDSA Public Key: Z - ECDSA Private Key: Z - HMAC Key: Z - PBKDF2 Derived Key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Z - RSA Public Key: Z - RSA Private Key: Z - User Password: Z

Table 13: Approved Services

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

- Generate: The module generates or derives the SSP.
- Read: The SSP is read from the module (e.g. the SSP is output).
- Write: The SSP is updated, imported, or written to the module.
- Execute: The module uses the SSP in performing a cryptographic operation.
- Zeroise: The module zeroises the SSP.

4.4 Non-Approved Services

N/A for this module.

None.

4.5 External Software/Firmware Loaded

None. The module does not use or require external software or firmware components.

5 Software/Firmware Security

5.1 Integrity Techniques

The module uses an RSA PKCS1.5 4096 with SHA2-256 (Cert. #A7626) signatures to verify the integrity of its binaries. The integrity check runs automatically when FortiClient starts. If the test passes, the results are output to the command prompt window.

If the test fails, FortiClient will log the failure and perform shutdown. See Section 10.4 (Error States) for more information.

5.2 Initiate on Demand

The integrity test can be triggered on demand by reinitializing the module or via the self-test commands.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

The FortiClient module operates in a modifiable operational environment and runs on a commercially available GPC. For specific information regarding the operational environment, please refer to section 2.3.1. There are no user-configurable settings for the module. For more details on the installation process, please refer to section 11.1.

7 Physical Security

N/A for this module.

As per section 7.7.1 of BS ISO/IEC 19790:2012, this section is not applicable due to this being a multi-chip embedded software only module.

8 Non-Invasive Security

This section is not applicable. There are no non-invasive security test's employed for this Module as per Annex F of BS ISO/IEC 19790:2012.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
SDRAM	System Memory as Plaintext	Dynamic

Table 14: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API Input	Calling application	SDRAM	Plaintext	Manual	Electronic	
API Output	SDRAM	Calling application	Plaintext	Manual	Electronic	

Table 15: SSP Input-Output Methods

9.3 SSP Zeroization Methods

As per SP IG 9.3 Resolution 2, Fortinet follows SSP procedural zeroization which is achieved through the following steps:

1. Uninstalling the Module
2. Overwriting the mass storage
3. Power Cycling the Host Computer

The operator must maintain control of the module during the entire zeroization procedure.

In accordance with IG 9.7.B Resolution 2, successful completion of the procedural zeroization itself serves as the implicit indicator that zeroization is complete.

Zeroization Method	Description	Rationale	Operator Initiation
Procedural Zeroisation	Clear volatile memory	All SSPs are stored in volatile memory and are cleared immediately when power is removed or the process is deleted.	Operator removes power from the module or uninstalls the module.

Zeroization Method	Description	Rationale	Operator Initiation
Zeroize function	Zeroization of SSPs in memory via a direct function call	Use of explicit zeroisation function destroys SSP information immediately by overwriting memory area with zeroes.	Calling application utilizes zeroize API commands.

Table 16: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES Key	Symmetric Key used to Encrypt and Decrypt Data	128, 192, 256 bits - 128, 192, 256 bits	Symmetric Key Cryptography - CSP	Symmetric Key Generation for AES		AES for Symmetric Encryption/Decryption
DH Private Key	Key Agreement and Establishment	2048 - 8192 bits - 112 - 200 bits	Asymmetric Private Key - CSP	KAS Key Pair Generation		FFC-SSC Safe Prime Key Generation
DH Public Key	Key Agreement and Establishment	2048 - 8192 bits - 112 - 200 bits	Asymmetric Public Key - PSP	KAS Key Pair Generation		FFC-SSC Safe Prime Key Generation
DH Shared Secret Key	Shared secret computation	2048 - 8192 bits - 112 - 200 bits	Shared Secret - CSP		FFC-SSC	FFC-SSC Safe Prime Key Generation
Entropy Input String	Input String from the Entropy Pool	256 bits - 256 bits	Entropy - CSP	Entropy Generation		DRBG
DRBG Seed	256 bit seed used by the DRBG	256 bits - 256 bits	Seeding Material for DRBG - CSP	DRBG		DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG 'V' Value	Internal State Value for the DRBG	256 bits - 256 bits	DRBG - CSP	DRBG		DRBG
DRBG 'Key' Value	Internal State Value for the DRBG	256 bits - 256 bits	DRBG - CSP	DRBG		DRBG
ECDH Private Key	Key Agreement and Establishment	P-256, P-384, P-521 - 128 - 256 bits	Asymmetric Private Key - CSP	KAS Key Pair Generation		ECC-SSC
ECDH Public Key	Key Agreement and Establishment	P-256, P-384, P-521 - 128 - 256 bits	Asymmetric Public Key - PSP	KAS Key Pair Generation		ECDSA for Key Verification ECC-SSC
ECDH Shared Secret Key	Shared secret computation	P-256, P-384, P-521 - 128 - 256 bits	Shared Secret - CSP		ECC-SSC	
ECDSA Public Key	Digital Signature verification	P-256, P-384, P-521 - 128 - 256 bits	Asymmetric Public Key - PSP	ECDSA for Key Generation		ECDSA for Signature Verification
ECDSA Private Key	Digital Signature Generation	P-256, P-384, P-521 - 128 - 256 bits	Asymmetric Private Key - CSP	ECDSA for Key Generation		ECDSA for Signature Generation
HMAC Key	Message Authentication	160, 256, 384, 512 - 112 - 256 bits	Message Authentication - CSP			HMAC for Keyed Hash Operation
PBKDF2 Derived Key	Key derived from PBKDF2	160 bits - 160 bits	Key Derivation - CSP	PBKDF		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
RSA Public Key	Asymmetric Key used for RSA signature verification	2048, 3072, 4096 - 112-152 bits	Asymmetric Public Key - PSP	RSA for Key Generation		RSA for Signature Verification
RSA Private Key	Asymmetric Key used for RSA signature generation	2048, 3072, 4096 - 112-152 bits	Asymmetric Private Key - CSP	RSA for Key Generation		RSA for Signature Generation
User Password	Input to PBKDF2 for Key Derivation	8 - 105 characters - 52 - 256	Key Derivation - CSP			HMAC for Keyed Hash Operation PBKDF

Table 17: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	
DH Private Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	DH Public Key:Paired With
DH Public Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	DH Private Key:Paired With
DH Shared Secret Key	API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	DH Public Key:Derived From DH Private Key:Derived From
Entropy Input String		SDRAM:Plaintext	Module lifetime	Procedural Zeroisation	
DRBG Seed		SDRAM:Plaintext	Module lifetime	Procedural Zeroisation	DRBG Entropy Input:Derived From
DRBG 'V' Value		SDRAM:Plaintext	Module lifetime	Procedural Zeroisation	DRBG Seed:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DRBG 'Key' Value		SDRAM:Plaintext	Module lifetime	Procedural Zeroisation	DRBG Seed:Derived From
ECDH Private Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	ECDH Public Key:Paired With
ECDH Public Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	ECDH Private Key:Paired With
ECDH Shared Secret Key	API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	ECDH Public Key:Derived From ECDH Private Key:Derived From
ECDSA Public Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	ECDSA Private Key:Paired With
ECDSA Private Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	ECDSA Public Key:Paired With
HMAC Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	
PBKDF2 Derived Key	API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	User Password:Derived From
RSA Public Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	RSA Private Key:Paired With
RSA Private Key	API Input API Output	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	RSA Public Key:Paired With
User Password	API Input	SDRAM:Plaintext	API Call Lifetime	Procedural Zeroisation	

Table 18: SSP Table 2

9.5 Transitions

The following transitions apply to the module and will be non-approved for cryptographic protection purposes after December 31, 2030:

- All use of the SHA-1 hash algorithm
- SSPs with less than 128-bit security strength

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256	Configuration Integrity Test	KAT	Critical Function	Pass Indicator Error Indicator	Verify
SHA2-256 and RSA SigVer	RSA PKCS1.5 4096 and SHA2-256	KAT	SW/FW Integrity	Pass Indicator Error Indicator	Sign and Verify

Table 19: Pre-Operational Self-Tests

The cryptographic module's pre-operational self-tests provide the operator with assurance that no faults have been introduced that could hinder correct operation, as outlined in ISO/IEC 19790:2012, section 7.10.1. Upon successfully passing the self-tests, the module logs a pass indicator message in cryptdlog.txt. Refer to section 10.6 for more information.

In the event that any of the self-tests fail, FortiClient logs an error indicator message for the specific test(s) in cryptdlog.txt. Refer to section 10.4 for examples and more information.

Configuration Integrity Test:
The module performs the startup configuration integrity test by performing the following steps:

- Upon first startup, fccryptd.exe calculates an HMAC using SHA2-256 for the configuration settings stored in the registry.
- The calculated HMAC value is then stored in the registry.
- When authorized configuration changes are made, fccryptd.exe recalculates and stores the new HMAC value.
- On subsequent startups, fccryptd.exe verifies the integrity by recalculating the HMAC and comparing it with the stored HMAC value.
- If the stored and recalculated HMAC values don't match, the configuration integrity test fails.

The configuration integrity test can be run on demand using fccrypt fips kat configuration at the command line. The following is an example of a successful self-test:

FIPS: HMAC self-test passed

SW/FW Integrity Test:
The module verifies the integrity of each file within the cryptographic boundary through RSA signature verification with SHA2-256. Each file is signed during development, and the public RSA key is delivered with the module binary.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC #1	Key Size:128	KAT	CAST	Pass Indicator Error Indicator	Encrypt	Initialization
AES-CBC #2	Key Size:128	KAT	CAST	Pass Indicator Error Indicator	Decrypt	Initialization
AES-ECB #1	Key Size:128	KAT	CAST	Pass Indicator Error Indicator	Encrypt	Initialization
AES-ECB #2	Key Size:128	KAT	CAST	Pass Indicator Error Indicator	Decrypt	Initialization
AES-GCM #1	Key Size:128	KAT	CAST	Pass Indicator Error Indicator	Encrypt	Initialization
AES-GCM #2	Key Size:128	KAT	CAST	Pass Indicator Error Indicator	Decrypt	Initialization
Counter DRBG #1	AES-CBC (256 bit)	Health Tests	CAST	Pass Indicator Error Indicator	Instantiate, Generate, and Reseed	Initialization
Counter DRBG #2	AES-CBC (256 bit)	KAT	CAST	Pass Indicator Error Indicator	Prediction Resistance: No Derivation Function: Yes	Initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA KeyGen (FIPS186-5)	P-256	PCT	PCT	Pass Indicator Error Indicator	Key Pair Generation	On key pair generation
ECDSA KeyVer (FIPS186-5)	P-256	PCT	PCT	Pass Indicator Error Indicator	Key Pair Verification	On key pair generation
ECDSA SigGen (FIPS186-5)	P-256	KAT	CAST	Pass Indicator Error Indicator	Sign	Initialization
ECDSA SigVer (FIPS186-5)	P-256	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
HMAC-SHA2-256	HMAC-SHA2-256	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
HMAC-SHA2-384	HMAC-SHA2-384	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
HMAC-SHA2-512	HMAC-SHA2-512	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
HMAC-SHA3-256	HMAC-SHA3-256	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
HMAC-SHA3-384	HMAC-SHA3-384	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
HMAC-SHA3-512	HMAC-SHA3-512	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
KAS-ECC-SSC	P-256	KAT	CAST	Pass Indicator	Verify computation of shared	Initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				Error Indicator	secret Z in Ephemeral Unified scheme	
KAS-FFC-SSC	MODP-2048	KAT	CAST	Pass Indicator Error Indicator	Verify computation of shared secret Z in dh Ephemeral scheme	Initialization
PBKDF2	HMAC-SHA-1	KAT	CAST	Pass Indicator Error Indicator	Key Derivation	Initialization
RSA KeyGen	2048 bit	KAT	CAST	Pass Indicator Error Indicator	Key Pair Generation	Initialization
RSA KeyGen PCT	2048 bit	PCT	PCT	Pass Indicator Error Indicator	Key Pair Generation	On key pair generation
RSA SigGen	2048 bit	KAT	CAST	Pass Indicator Error Indicator	Sign	Initialization
RSA SigVer	2048 bit	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
SHA2-256	SHA2-256	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
SHA2-384	SHA2-384	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
SHA2-512	SHA2-512	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA3-256	SHA3-256	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
SHA3-384	SHA3-384	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
SHA3-512	SHA3-512	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
SHA3-256 (A4973)	SHA3-256	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization
Safe Primes Key Generation (A7626)	Safe Prime Groups: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	PCT	PCT	Pass Indicator Error Indicator	Key Pair Generation	On key pair generation
HMAC-SHA-1 (A7626)	HMAC-SHA-1	KAT	CAST	Pass Indicator Error Indicator	Verify	Initialization

Table 20: Conditional Self-Tests

Cryptographic Algorithm Self Tests (CASTs): The module performs self-tests on approved algorithms, as required by FIPS 140-3 Implementation Guidance (IG) section 10.3.A. If any self-test fails, FortiClient logs the specific test(s) in cryptdlog.txt (refer to Section 10.6 for more information).

The module also performs Critical Functions Test (CFT), Continuous Pair-wise Consistency Test (CPCT), as outlined in FIPS 140-3 IG section 10.3.A.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256	KAT	Critical Function	On-Demand	Initialization
SHA2-256 and RSA SigVer	KAT	SW/FW Integrity	On-Demand	Initialization

Table 21: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC #1	KAT	CAST	On-Demand	Initialization
AES-CBC #2	KAT	CAST	On-Demand	Initialization
AES-ECB #1	KAT	CAST	On-Demand	Initialization
AES-ECB #2	KAT	CAST	On-Demand	Initialization
AES-GCM #1	KAT	CAST	On-Demand	Initialization
AES-GCM #2	KAT	CAST	On-Demand	Initialization
Counter DRBG #1	Health Tests	CAST	On-Demand	Initialization
Counter DRBG #2	KAT	CAST	On-Demand	Initialization
ECDSA KeyGen (FIPS186-5)	PCT	PCT	On-Demand	On key pair generation
ECDSA KeyVer (FIPS186-5)	PCT	PCT	On-Demand	On key pair generation
ECDSA SigGen (FIPS186-5)	KAT	CAST	On-Demand	Initialization
ECDSA SigVer (FIPS186-5)	KAT	CAST	On-Demand	Initialization
HMAC-SHA2-256	KAT	CAST	On-Demand	Initialization
HMAC-SHA2-384	KAT	CAST	On-Demand	Initialization
HMAC-SHA2-512	KAT	CAST	On-Demand	Initialization
HMAC-SHA3-256	KAT	CAST	On-Demand	Initialization
HMAC-SHA3-384	KAT	CAST	On-Demand	Initialization

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA3-512	KAT	CAST	On-Demand	Initialization
KAS-ECC-SSC	KAT	CAST	On-Demand	Initialization
KAS-FFC-SSC	KAT	CAST	On-Demand	Initialization
PBKDF2	KAT	CAST	On-Demand	Initialization
RSA KeyGen	KAT	CAST	On-Demand	Initialization
RSA KeyGen PCT	PCT	PCT	On-Demand	On key pair generation
RSA SigGen	KAT	CAST	On-Demand	Initialization
RSA SigVer	KAT	CAST	On-Demand	Initialization
SHA2-256	KAT	CAST	On-Demand	Initialization
SHA2-384	KAT	CAST	On-Demand	Initialization
SHA2-512	KAT	CAST	On-Demand	Initialization
SHA3-256	KAT	CAST	On-Demand	Initialization
SHA3-384	KAT	CAST	On-Demand	Initialization
SHA3-512	KAT	CAST	On-Demand	Initialization
SHA3-256 (A4973)	KAT	CAST	On-Demand	Initialization
Safe Primes Key Generation (A7626)	PCT	PCT	On-Demand	On key pair generation
HMAC-SHA-1 (A7626)	KAT	CAST	On-Demand	Initialization

Table 22: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Critical Error State	FortiClient Logs the Specific Test or Tests that failed in cryptdlog.txt	Boot Up Manual	Power-Cycle	Check Logs 2025-08-13 01:08:03.763 [4824, 2300] - CryptdFipsEnterErrorMode()--> msg = FIPS: Running FIPS self-test ... FIPS Error: AES-CBC self-test failed. FIPS Error: Running FIPS self-test --> failed.

Table 23: Error States

10.5 Operator Initiation of Self-Tests

The startup self-tests can also be initiated on demand from the Windows command prompt using the following commands:

- `FCCryptd.exe fips kat all` (to initiate all self-tests)
- `FCCryptd.exe fips kat <test>` (to initiate a specific self-test)
- `FCCryptd.exe fips <test>` (to list available tests)

10.6 Additional Information

The results of the self-tests are logged to a file. For a default installation of FortiClient on the C: drive, you can view the Cryptdlog.txt file in: C:\Program Files\Fortinet\FortiClient\logs\fips.

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

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The FortiClient FIPS 140-3 installer packages are installed as a standalone crypto module alongside a FortiClient installation. Operating the module without following the guidance in Section 11 will result in non-compliant behavior and is outside the scope of this Security Policy.

The basic steps in the deployment process are as follows:

1. Extract the FortiClient zip file FortiClientSetup_7.4.4.8927_x64.zip
2. Open a terminal with Administrator privileges in the extracted directory and run the following command:
   ```
   msiexec /i forticlient.msi NO_FEATURE_SELECT=1
   ADDLOCAL=Feature_Basic,Feature_Core,Feature_EndPointNAC,Feature_SSLVPN,
   Feature_VPN,Feature_Vulnerability,Feature_FIPS
   ```
3. The installer window should open. Follow the instructions to successfully install the FortiClient software.
4. During the process, ensure that the installation directory is set to C:/Program Files/Fortinet/FortiClient

It is important to note that the version number on the zip files includes the build number. Thus, to verify the installed version, open a Windows Command Prompt window as an administrator and run the following command:

- `fccryptd.exe fips version`

The command runs the software integrity test and displays the module version as shown in the following image.

```
> .\FCCryptd.exe fips version

FIPS: Integrity test passed (FCCryptd.exe).
FIPS: Integrity test passed (FCCryptDLL.dll).
FIPS: Integrity test passed (libcrypto-3-x64.dll).
FIPS: Integrity test passed (libssl-3-x64.dll).
FIPS: Integrity test passed (fips.dll).
FIPS: Integrity test passed (fipsmodule.cnf).
FortiClient Cryptographic Module 1.0
```

Launch the FortiClient application as you would launch any other Windows application. Failure to perform the steps outlined in this section will result in the module operating in a non-compliant state.

11.2 Administrator Guidance

FortiClient administrator guidance is publicly available from the Fortinet Technical Documentation site. The key administrator guidance documents are listed below:

- FortiClient Administration Guide 7.4
- FortiClient Administration Guide 8.0

11.3 Non-Administrator Guidance

All guidance is included in the FortiClient Administration Guide.

12 Mitigation of Other Attacks

Not Applicable.