

**Versa Networks Controller v1.0  
by Versa Networks, Inc.**

**FIPS 140-2 Level 1 Non-Proprietary Security  
Policy**

**Document Version Number: 1.2**

**Date: November 18, 2022**

## Table of Contents

1. Module Overview .....	3
2. Modes of Operation .....	5
2.1 Approved and Allowed Cryptographic Functions .....	6
2.2 Non-approved algorithms allowed in FIPS-approved mode .....	9
2.3 All other algorithms.....	9
3. Ports and interfaces .....	10
4. Roles and Services .....	11
5. Cryptographic Keys and CSPs .....	13
6. Self-tests .....	15
7. References.....	17

# 1. Module Overview

The Versa Networks Controller (aka Versa Controller, which is the cryptographic module) plays a key role in the Versa solution, providing a control plane entry point for Versa Networks Branch deployments. The Versa Controller performs the following tasks:

Establish and maintain a secure control channel with each remote Versa Networks Branch — The secure channel, established with the Controller using IPsec, provides the transport mechanism for exchange of control information between branch appliances. The Versa Controller serves as an attachment point for management and control purposes between Versa management components (Versa Director and Versa Analytics), and the Versa Networks Branch modules.

At least one Versa Controller is required per Versa SD-WAN deployment. Multiple Versa Controllers can be deployed in a Versa SD-WAN deployment to provide high availability.

The cryptographic module is a software module that is executing in a modifiable operational environment by a general-purpose computer.

For the purposes of the FIPS 140-2 validation, the module is a software-only, multi-chip standalone cryptographic module. FIPS 140-2 conformance testing was performed at Security Level 1. The following configuration was tested by the lab.

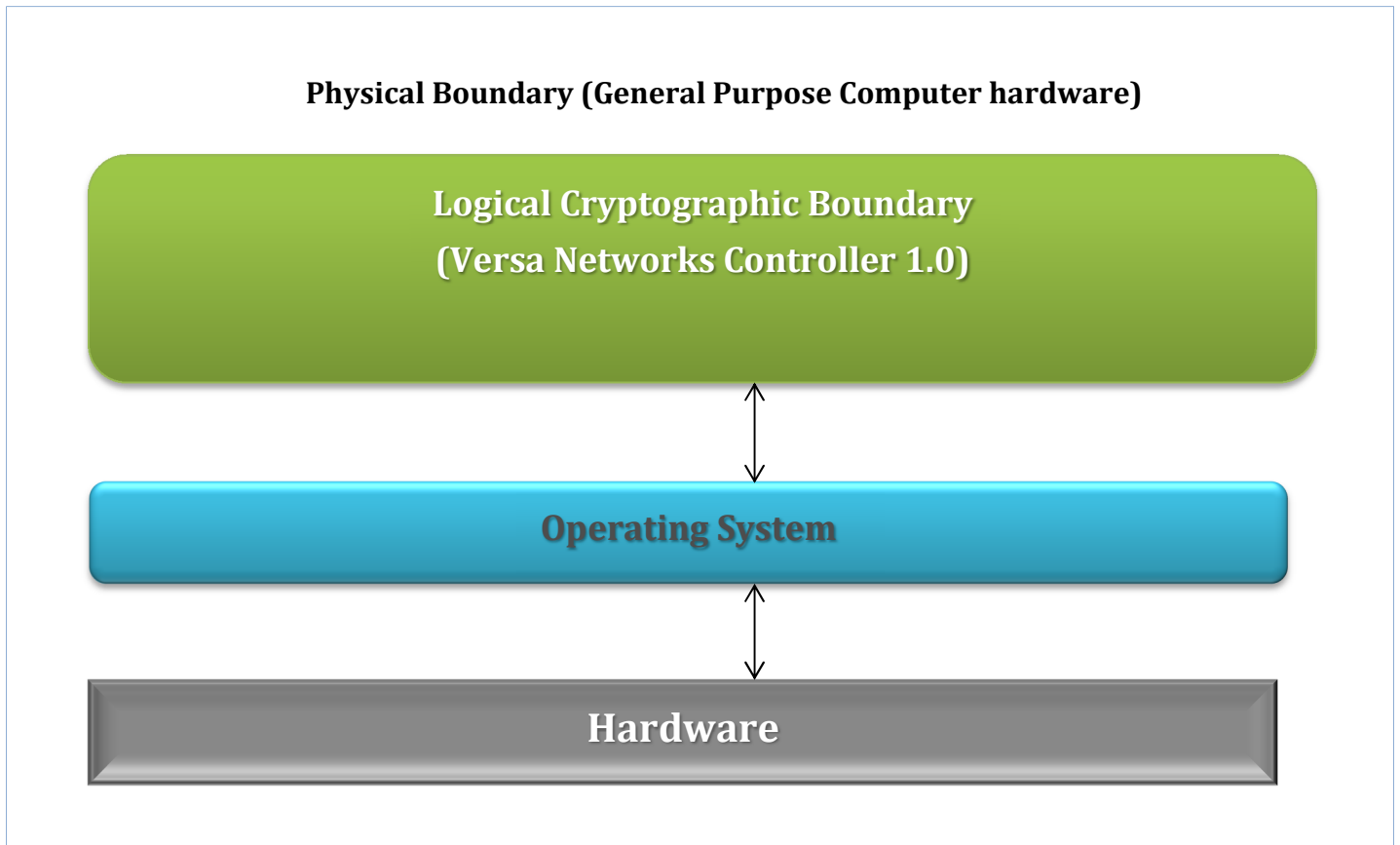
Table 1.1: Configuration tested by the lab

Product	Platform	Processors	Operating Systems
Versa Networks Controller	Versa CSG770	Intel Denverton C3708 with and without AES-NI	Ubuntu 18.04

Table 1.2: Module Security Level Statement

FIPS Security Area	Security Level
Cryptographic Module Specification	1
Module Ports and Interfaces	1
Roles, Services and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Figure 1: Block Diagram for the product



## 2. Modes of Operation

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

FIPS Mode (Approved mode of operation): Only approved or allowed security functions with sufficient security strength can be used. To enable FIPS mode, all existing configuration shall be erased, before running the request command to enable FIPS mode on the Versa Operational CLI. After enabling the FIPS mode, Versa services need to be restarted to operate the module in FIPS mode.

```
admin> request system fips-mode enable
Please Erase config. Config Erased? Are you sure? [no,yes]yes
Restart all Versa services. Are you sure? [no,yes]yes

admin>
```

Non-FIPS Mode (non-Approved mode of operation): All approved and non-approved security functions can be used. To disable FIPS mode, all existing configuration shall be erased, before running the request command to disable FIPS mode on the Versa Operational CLI. After disabling the FIPS mode, Versa services must be restarted to operate the module in non-FIPS mode.

```
admin> request system fips-mode disable
Please Erase config. Config Erased? Are you sure? [no,yes]yes
Restart all Versa services. Are you sure? [no,yes]yes

admin>
```

## 2.1 Approved and Allowed Cryptographic Functions

The following approved cryptographic algorithms are used in FIPS approved mode of operation.

Table 2.1: Approved Cryptographic Functions.

CAVP Cert	Library	Algorithm	Standard	Model/Method	Key Lengths, Curves or Moduli	Use
<u>C1971</u>	VOS™ IPsec Cryptographic Module	CVL IKEv1 KDF <sup>2</sup> CVL IKEv2 KDF <sup>2</sup>	SP800-135			IPsec Key Derivation
<u>C1971</u>	VOS™ IPsec Cryptographic Module	AES	FIPS197, SP800-38A, SP800-38F	CBC, ECB, CTR	128,256	IPsec Symmetric encryption, decryption KTS <sup>3</sup>
<u>A1644</u>	VOS™ IPsec Cryptographic Module	AES	FIPS197, SP800-38A, SP800-38F	CBC, CTR, ECB	128,256	
<u>C1971</u>	VOS™ IPsec Cryptographic Module	HMAC	FIPS198-1	HMAC-SHA-1	160-bit	IPsec Message Authentication Code
				HMAC-SHA-224	224-bit	
				HMAC-SHA-256	256-bit	
				HMAC-SHA-384	384-bit	
				HMAC-SHA-512	512-bit	
<u>A1644</u>	VOS™ IPsec Cryptographic Module	KAS-ECC-SSC	SP800-56Ar3	ECC Ephemeral Unified Scheme	P-224, P-256, P-384 corresponding to 112 - 192 bits of security	IPsec Shared Secret Computation
<u>C1971</u>	VOS™ IPsec Cryptographic Module	RSA	FIPS186-4	PKCS#1v1.5 with SHA-256, SHA-384, SHA-512	1024, 2048, 3072	IPsec Digital Signature Verification
<u>C1971</u>	VOS™ IPsec Cryptographic Module	RSA	FIPS186-4	PSS with SHA-1, SHA-256, SHA-384, SHA-512	1024, 2048, 3072	IPsec Digital Signature Verification
<u>C1971</u>	VOS™ IPsec Cryptographic Module	SHS	FIPS180-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512		IPsec Message Digest
<u>C1923</u>	VOS™ TLS Cryptographic Module	AES	FIPS197, SP800-38A	CBC, ECB, CTR	128, 192, 256	SSHv2 Symmetric encryption, decryption
<u>A1636</u>	VOS™ TLS Cryptographic Module	AES	FIPS197, SP800-38A	CBC, ECB, CTR	128, 192, 256	SSHv2 Symmetric encryption, decryption

CAVP Cert	Library	Algorithm	Standard	Model/Method	Key Lengths, Curves or Moduli	Use
<u>C1923</u>	VOS™ TLS Cryptographic Module	DRBG	SP800-90A	CTR_DRBG with AES-256		IPsec, SSHv2 Deterministic Random Bit Generation
<u>C1923</u>	VOS™ TLS Cryptographic Module	ECDSA	FIPS186-4		P-256, P-384, P-521	SSHv2 Key Pair Generation
<u>C1923</u>	VOS™ TLS Cryptographic Module	ECDSA	FIPS186-4	SHA-224, SHA-256, SHA-384, SHA-512	P-256, P-384, P-521	SSHv2 Digital Signature Generation
<u>C1923</u>	VOS™ TLS Cryptographic Module	ECDSA	FIPS186-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	P-256, P-384, P-521	SSHv2 Digital Signature Verification
<u>C1923</u>	VOS™ TLS Cryptographic Module	HMAC	FIPS198-1	HMAC-SHA-1	160-bit	SSHv2 Message Authentication Code
				HMAC-SHA-224	224-bit	
				HMAC-SHA-256	256-bit	
				HMAC-SHA-384	384-bit	
				HMAC-SHA-512	512-bit	
<u>A1636</u>	VOS™ TLS Cryptographic Module	KAS-ECC-SSC	SP800-56Ar3	ECC Ephemeral Unified Scheme	P-256, P-384, P-521 corresponding to 128 - 256 bits of security	SSHv2 Shared Secret Computation
<u>C1923</u>	VOS™ TLS Cryptographic Module	KDF CVL SSHv2 <sup>2</sup>	SP800-135			SSHv2 Key Derivation
<u>C1923</u>	VOS™ TLS Cryptographic Module	RSA	FIPS186-4		2048, 3072	IPsec, SSHv2 Key Pair Generation
<u>C1923</u>	VOS™ TLS Cryptographic Module	RSA	FIPS186-4	PKCS#1v1.5 with SHA-224, SHA-256, SHA-384, SHA-512	2048, 3072	IPsec, SSHv2 Digital Signature Generation
<u>C1923</u>	VOS™ TLS Cryptographic Module	RSA	FIPS186-2	PKCS#1v1.5 with SHA1, SHA-224, SHA-256, SHA-384, SHA-512	1024, 1536, 2048, 3072, 4096	SSHv2 Digital Signature Verification
<u>C1923</u>	VOS™ TLS Cryptographic Module	SHS	FIPS180-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512		SSHv2 Message Digest

CAVP Cert	Library	Algorithm	Standard	Model/Method	Key Lengths, Curves or Moduli	Use
<u>A1644</u> <u>C1971</u>	VOS™ IPsec Cryptographic Module	KAS	SP800-56Ar3 and SP800-135	ECC Ephemeral Unified Scheme	P-224, P-256, P-384	IPsec Shared Secret Computation IPsec Key Derivation
<u>A1636</u> <u>C1923</u>	VOS™ TLS Cryptographic Module	KAS	SP800-56Ar3 and SP800-135	ECC Ephemeral Unified Scheme	P-256, P-384, P-521	SSHv2 Shared Secret Computation TLS, SSHv2 Key Derivation
CKG (Vendor Affirmed)		Cryptographic Key Generation	SP800-133			Key Generation <sup>1</sup>

Note 1: not all CAVS tested modes of the algorithms are used in this module.

<sup>1</sup> CKG is only used to generate asymmetric keys. The module directly uses the output of the DRBG. The generated seed used in the asymmetric key generation is an unmodified output from DRBG. Section 4, example 1, of SP800-133r2 “Using the Output of a Random Bit Generator” is applicable.

<sup>2</sup>No parts of these protocols, other than the KDF, have been tested by the CAVP and CMVP.

<sup>3</sup> KTS: AES-CBC and CTR (Certs. #A1644 and #C1971) and HMAC (Cert. #C1971); key establishment methodology provides 128 or 256 bits of encryption strength.



## 2.2 Non-approved algorithms allowed in FIPS-approved mode

In FIPS approved mode of operation, only FIPS-approved and certified algorithms are allowed. In FIPS approved mode, none of the non-FIPS-approved or non-certified algorithms are allowed to be used.

## 2.3 All other algorithms

The below table indicates non-FIPS-approved algorithms that are only permitted to be used in non-FIPS-approved mode.

**Table 2.3: Non-Approved Cryptographic Functions**

Algorithm	Use
RSA signature generation with keys of size < 2048 bits	Digital Signature Generation
RSA encryption	Key wrapping
3DES (non-compliant) MD5	IPSEC
ECDSA with curves P-192, K-163, B-163 and non-NIST curves KeyGen, SigGen	Digital Signature
ECDH with curves B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571 and non-NIST curves	Key Establishment – Non-compliant
HMAC with less than 112 bits key	Keyed Hash

### 3. Ports and interfaces

The physical ports of the module are the same as those of the computer system on which it is executing. The logical interfaces of the module are implemented via an Application Programming Interface (API). The table below describes the logical interfaces provided by the module.

**Table 3: FIPS 140-2 Logical Interfaces.**

<b>Logical Interface</b>	<b>Description</b>
Data Input	Input parameters that are supplied to the API commands
Data Output	Output parameters that are returned by the API commands
Control Input	API commands
Status Output	Return status provided by API commands

## 4. Roles and Services

### Roles

The module supports the following roles:

- **User Role:** performs services of establish, maintain, and close SSH and IPsec sessions.
- **Crypto Officer Role:** performs services of module installation and configuration of Versa services, monitoring status, troubleshooting, and running self-tests.

### Services

The module provides services to users that assume one of the available roles. All services are shown in Table 4 and described in detail in the user documentation.

Table 4 shows the services, the roles to perform the service, the cryptographic keys, or Critical Security Parameters (CSPs) involved and how they are accessed. The following convention is used to specify access rights to a CSP:

- **Create:** create a new CSP.
- **Read:** read the CSP.
- **Update:** write a new value to the CSP.
- **Zeroize:** zeroize the CSP.
- **Blank:** does not access any CSP or key during its operation.

**Table 4: Roles and Services**

Service	Corresponding Roles	Types of Access to Cryptographic Keys and CSPs
Module Installation	Crypto Officer	
Enable/Disable FIPS Mode	Crypto Officer	<b>All Keys/Certificates:</b> Zeroize
Configure SSH, IPsec, and perform general configuration	Crypto Officer	<b>SSH Keys:</b> Create, Read, Update, Delete <b>IPsec Keys, Certificates:</b> Create, Read, Update, Delete <b>DRBG CSPs:</b> Read, Update
Show status	Crypto Officer	<b>SSH Keys:</b> Create, Read <b>IPsec Keys, Certificates:</b> Create, Read <b>DRBG CSPs:</b> Read, Update
Self-Tests	Crypto Officer	
SSH Connection	Crypto Officer	<b>SSH Keys:</b> Create, Read <b>DRBG CSPs:</b> Read, Update
Zeroization	Crypto Officer	<b>All Keys/Certificates:</b> Zeroize
Reboot	Crypto Officer	
IPsec Connection to Versa Director (management platform)	User	<b>IPsec Keys, Certificates:</b> Create, Read <b>DRBG CSPs:</b> Read, Update

Service	Corresponding Roles	Types of Access to Cryptographic Keys and CSPs
IPsec Connection to Versa Analytics (Log/Audit event collector)	User	<b>IPsec Keys, Certificates:</b> Create, Read <b>DRBG CSPs:</b> Read, Update
IPsec Connection to Versa Networks Branch modules	User	<b>IPsec Keys, Certificates:</b> Create, Read <b>DRBG CSPs:</b> Read, Update

## 5. Cryptographic Keys and CSPs

The table below describes cryptographic keys and CSPs used by the module.

**Table 5: Cryptographic Keys and CSPs**

Protocol	Key	Description/Usage	Storage
IPsec	RSA private keys Established using DRBG	Used in IPsec handshake.	DRAM (plaintext)
IPsec	RSA public keys Established using DRBG	Used in IPsec handshake.	DRAM (plaintext)
IPsec	Elliptic Curve Diffie Hellman private key Established using DRBG	The private key used in Diffie Hellman (DH) exchange.	DRAM (plaintext)
IPsec	Elliptic Curve Diffie Hellman public key Established using DRBG	The public key used in Diffie Hellman (DH) exchange. This key is derived per the Diffie-Hellman key agreement.	DRAM (plaintext)
IPsec	Pre-Shared Key Set by operators	A shared secret known only to IPsec peers.	DRAM (plaintext)
IPsec	AES Encryption Keys Established using KDF IKE	The IPsec encryption keys.	DRAM (plaintext)
IPsec	HMAC Keys Established using KDF IKE	Used for computation of Hash-based Message Authentication Code (HMAC).	DRAM (plaintext)
SSHv2	AES Encryption Keys Established using KDF SSH	These are the SSHv2 session keys. They are used to encrypt all SSHv2 data traffic traversing between the endpoints.	DRAM (plaintext)
SSHv2	HMAC Keys Established using KDF SSH	Used for computation of Hash-based Message Authentication Code (HMAC).	DRAM (plaintext)
SSHv2	Elliptic Curve Diffie Hellman private key Established using DRBG	The private key used in Diffie Hellman (DH) exchange.	DRAM (plaintext)
SSHv2	Elliptic Curve Diffie Hellman public key Established using DRBG	The public key used in Diffie Hellman (DH) exchange. This key is derived per the Diffie-Hellman key agreement.	DRAM (plaintext)

Protocol	Key	Description/Usage	Storage
SSHv2	ECDSA private key Established using DRBG	Used in SSHv2 handshake.	DRAM (plaintext)
SSHv2	ECDSA public key Established using DRBG	Used in SSHv2 handshake.	DRAM (plaintext)
IPsec, SSHv2	CTR_DRBG CSPs: entropy input, V and Key  Entropy is loaded externally	Used during generation of random numbers.	DRAM (plaintext)

Note-1: public keys are not considered CSPs

Note-2: All keys, that are generated by this module, are generated by using AES-CTR-DRBG with AES-256. Entropy is loaded externally. Minimum number of bits of entropy loaded is 256-bits, since the minimum length of the entropy field is 256-bits.

Note-3: Keys can be provided to the module via API input parameters and output via API output parameters. The module does not enter or output keys outside its physical boundary. Zeroization is performed using power cycle.

## 6. Self-tests

The module performs the following power-up and conditional self-tests. Upon failure or a power-up or conditional self-test the module halts its operation.

### Power-Up Tests

The module performs power-up tests when the module is restarted or when the underlying system is rebooted or power-cycled, without operator intervention. Power-up tests ensure that the module is not corrupted and that the cryptographic algorithms work as expected.

While the module is executing the power-up tests, services are not available, and input and output are inhibited. The module is not available for use until the power-up tests are completed successfully.

If any power-up test fails, the module disables all services, and then enters error state. Thus, no further cryptographic operations are possible. If the power-up tests complete successfully, the module will startup normally and will provide the cryptographic operation service requests.

### Conditional Tests

The module performs conditional tests on the cryptographic algorithms, using the Pair-wise Consistency Tests (PCT) and Continuous Random Number Generator Test (CRNGT).

The software integrity test is performed during startup using RSA 2048-bit keys, with SHA256 Signature Verification.

**Table 6: Self-Tests**

Algorithm	Test
Software Integrity Power On Self Test	Software integrity test is performed using RSA key of length of at least 2048 bits and SHA256 Signature Verification
Power On Self Test for VOS TLS Cryptographic Subsystem	For VOS TLS Cryptographic Subsystem, the test validation routines are performed: <ol style="list-style-type: none"><li>AES encryption and decryption are separately tested (key sizes tested: 128, 192, 256)<ol style="list-style-type: none"><li>CBC</li><li>ECB</li><li>CTR</li></ol></li><li>RSA KAT (key sizes tested: 2048, 3072), using SHA256</li><li>SHA<ol style="list-style-type: none"><li>SHA-1</li><li>SHA-224</li><li>SHA-256</li><li>SHA-384</li><li>SHA-512</li></ol></li></ol>

Algorithm	Test
	<ol style="list-style-type: none"> <li>4. HMAC               <ol style="list-style-type: none"> <li>a. HMAC-SHA-1</li> <li>b. HMAC-SHA-224</li> <li>c. HMAC-SHA-256</li> <li>d. HMAC-SHA-384</li> <li>e. HMAC-SHA-512</li> </ol> </li> <li>5. DRBG KAT</li> <li>6. KAS (ECC-SSC) Primitive "Z" Computation KAT per implementation guidance</li> <li>7. ECDSA PCT (curve sizes P-224, K-233) using SHA512</li> <li>8. KDF KAT – For SSH</li> </ol>
Power On Self Test for IPsec Subsystem	<p>For IPsec Subsystem, the test validation routines are performed:</p> <ol style="list-style-type: none"> <li>1. AES encryption and decryption are separately tested (key sizes tested are 128, 192, 256)               <ol style="list-style-type: none"> <li>a. CBC</li> <li>b. ECB</li> </ol> </li> <li>2. RSA KAT (key sizes tested are 2048, 3072)</li> <li>3. SHA               <ol style="list-style-type: none"> <li>a. SHA-1</li> <li>b. SHA-224</li> <li>c. SHA-256</li> <li>d. SHA-384</li> <li>e. SHA-512</li> </ol> </li> <li>4. HMAC               <ol style="list-style-type: none"> <li>a. HMAC-SHA-1</li> <li>b. HMAC-SHA-224</li> <li>c. HMAC-SHA-256</li> <li>d. HMAC-SHA-384</li> <li>e. HMAC-SHA-512</li> </ol> </li> <li>5. KAS (ECC-SSC) Primitive "Z" Computation KAT per implementation guidance</li> <li>6. KDF KAT – For IKEv1 and IKEv2</li> </ol>
Conditional Test for RSA Key Generation	PCT using SHA-256, signature generation and verification. PCT for encryption and decryption.
Conditional Test for ECDSA Key Generation	PCT using SHA-256, signature generation and verification.
Conditional Test for ECC DH Public/Private Key Validation (IPsec)	ECC DH Private/Public Key Validation tests as per SP800-56Ar3 including ECC Full Public-Key Validation Routine
Conditional Test for ECC DH Public/Private Key Validation (SSH)	ECC DH Private/Public Key Validation tests as per SP800-56Ar3 including ECC Full Public-Key Validation Routine
DRBG	DRBG Health Test Performed per SP 800-90A Section 11.3
Conditional Test for DRBG	CRNGT
Entropy Test	Continuous Random Number Test for the entropy data



## 7. References

**Table 7: References**

Reference	Specification
[ANS X9.31]	Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)
[FIPS 140-2]	Security Requirements for Cryptographic modules, May 25, 2001
[FIPS 180-4]	Secure Hash Standard (SHS)
[FIPS 186-2/4]	Digital Signature Standard
[FIPS 197]	Advanced Encryption Standard
[FIPS 198-1]	The Keyed-Hash Message Authentication Code (HMAC)
[FIPS 202]	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions
[PKCS#1 v2.1]	RSA Cryptography Standard
[PKCS#5]	Password-Based Cryptography Standard
[PKCS#12]	Personal Information Exchange Syntax Standard
[SP 800-38A]	Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode
[SP 800-38B]	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication
[SP 800-38C]	Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality
[SP 800-38F]	Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping
[SP 800-56A]	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography
[SP 800-56B]	Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography
[SP 800-56C]	Recommendation for Key Derivation through Extraction-then-Expansion
[SP 800-67R1]	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
[SP 800-89]	Recommendation for Obtaining Assurances for Digital Signature Applications
[SP 800-90A]	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
[SP 800-108]	Recommendation for Key Derivation Using Pseudorandom Functions
[SP 800-132]	Recommendation for Password-Based Key Derivation
[SP 800-135]	Recommendation for Existing Application –Specific Key Derivation Functions