

Red Hat(R), Inc.

Red Hat Enterprise Linux 9 - OpenSSL FIPS Provider

FIPS 140-3 Non-Proprietary Security Policy

Document Version: 1.1

Last Modified: 2026/11/09

Prepared by:

atsec information security corporation

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

www.atsec.com

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Table of Contents

1 General 6
1.1 Overview 6
1.1.1 How this Security Policy was prepared 6
1.2 Security Levels 6
2 Cryptographic Module Specification 8
2.1 Description 8
2.2 Tested and Vendor Affirmed Module Version and Identification 9
2.3 Excluded Components 10
2.4 Modes of Operation 10
2.5 Algorithms 11
2.6 Security Function Implementations 25
2.7 Algorithm Specific Information 35
2.7.1 AES GCM IV 35
2.7.2 AES XTS 35
2.7.3 Key Derivation using SP 800-132 PBKDF2 36
2.7.4 SP 800-56A Rev. 3 Assurances 36
2.7.5 SHA-3 36
2.7.6 RSA Signatures 37
2.7.7 Key Agreement 37
2.7.8 Key Transport 37
2.8 RBG and Entropy 37
2.9 Key Generation 38
2.10 Key Establishment 38
2.11 Industry Protocols 39
3 Cryptographic Module Interfaces 40
3.1 Ports and Interfaces 40
4 Roles, Services, and Authentication 41
4.1 Authentication Methods 41
4.2 Roles 41
4.3 Approved Services 41

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

4.4 Non-Approved Services 55
4.5 External Software/Firmware Loaded 56
5 Software/Firmware Security 57
5.1 Integrity Techniques 57
5.2 Initiate on Demand 57
6 Operational Environment 58
6.1 Operational Environment Type and Requirements 58
6.2 Configuration Settings and Restrictions 58
6.3 Additional Information 58
7 Physical Security 59
8 Non-Invasive Security 60
9 Sensitive Security Parameters Management 61
9.1 Storage Areas 61
9.2 SSP Input-Output Methods 61
9.3 SSP Zeroization Methods 61
9.4 SSPs 62
9.5 Transitions 76
10 Self-Tests 77
10.1 Pre-Operational Self-Tests 77
10.2 Conditional Self-Tests 77
10.3 Periodic Self-Test Information 80
10.4 Error States 82
10.5 Operator Initiation of Self-Tests 82
11 Life-Cycle Assurance 83
11.1 Installation, Initialization, and Startup Procedures 83
11.2 Administrator Guidance 83
11.3 Non-Administrator Guidance 83
11.4 Design and Rules 83
11.5 Maintenance Requirements 83
11.6 End of Life 83
12 Mitigation of Other Attacks 84
Appendix A. Glossary and Abbreviations 85
Appendix B. References 87

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

List of Tables

Table 1: Security Levels 7
Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets) 9
Table 3: Tested Operational Environments - Software, Firmware, Hybrid 10
Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid 10
Table 5: Modes List and Description 11
Table 6: Approved Algorithms 23
Table 7: Vendor-Affirmed Algorithms 23
Table 8: Non-Approved, Not Allowed Algorithms 24
Table 9: Security Function Implementations 35
Table 10: Entropy Certificates 37
Table 11: Entropy Sources 37
Table 12: Ports and Interfaces 40
Table 13: Roles 41
Table 14: Approved Services 55
Table 15: Non-Approved Services 56
Table 16: Storage Areas 61
Table 17: SSP Input-Output Methods 61
Table 18: SSP Zeroization Methods 62
Table 19: SSP Table 1 70
Table 20: SSP Table 2 76
Table 21: Pre-Operational Self-Tests 77
Table 22: Conditional Self-Tests 80
Table 23: Pre-Operational Periodic Information 80
Table 24: Conditional Periodic Information 81
Table 25: Error States 82

List of Figures

Figure 1: Block Diagram 9

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for version 3.0.7-cda111b5812c30d4 of the Red Hat Enterprise Linux 9 - OpenSSL FIPS Provider. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 1 module.

This Non-Proprietary Security Policy may be reproduced and distributed, but only whole and intact and including this notice. Other documentation is proprietary to their authors.

1.1.1 How this Security Policy was prepared

In preparing the Security Policy document, the laboratory formatted the vendor-supplied documentation for consolidation without altering the technical statements therein contained. The further refining of the Security Policy document was conducted iteratively throughout the conformance testing, wherein the Security Policy was submitted to the vendor, who would then edit, modify, and add technical contents. The vendor would also supply additional documentation, which the laboratory formatted into the existing Security Policy, and resubmitted to the vendor for their final editing.

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	1

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Section	Title	Security Level
	Overall Level	1

Table 1: Security Levels

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Red Hat Enterprise Linux 9 - OpenSSL FIPS Provider (hereafter referred to as "the module") is defined as a software module in a multi-chip standalone embodiment. It provides a C language application program interface (API) for use by other applications that require cryptographic functionality. The module consists of one software component, the "FIPS provider", which implements the FIPS requirements and the cryptographic functionality provided to the operator.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The cryptographic boundary of the module is defined as the fips.so shared library, which contains the compiled code implementing the FIPS provider.

Tested Operational Environment's Physical Perimeter (TOEPP):

The TOEPP of the module is defined as the general-purpose computer on which the module is installed.

Figure 1 shows a block diagram that represents the design of the module when the module is operational and providing services to other user space applications. In this diagram, the physical perimeter of the operational environment (a general-purpose computer on which the module is installed) is indicated by a purple dashed line. The cryptographic boundary is represented by the component painted in orange block, which consists only of the shared library implementing the FIPS provider (fips.so).

Green lines indicate the flow of data between the cryptographic module and its operator application, through the logical interfaces defined in Section 3.

Components in white are only included in the diagram for informational purposes. They are not in-cluded in the cryptographic boundary (and therefore not part of the module's validation). For exam-ple, the kernel is responsible for managing system calls issued by the module itself, as well as other applications using the module for cryptographic services.

The PAA/PAI provided by the processor is located within the module's physical perimeter but outside of the module's cryptographic boundary.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

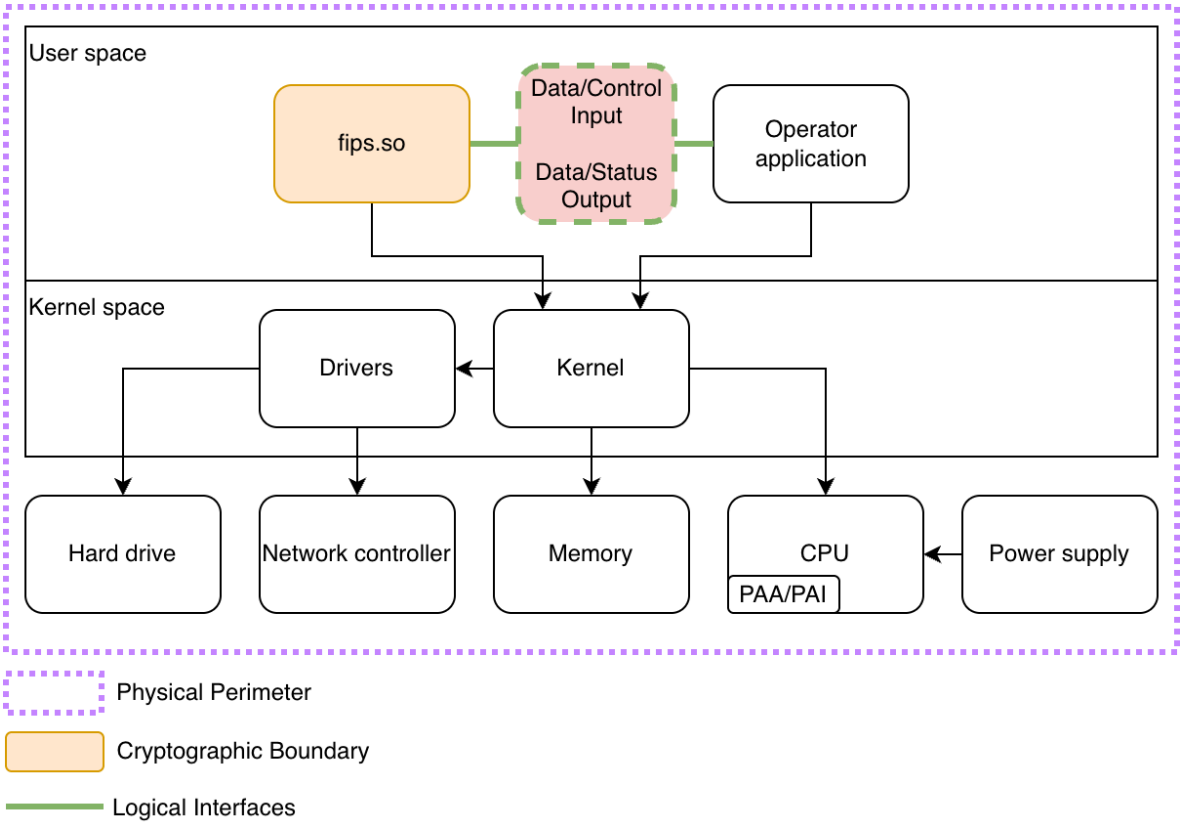


Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
fips.so on Red Hat Enterprise Linux 9 and Intel(R) Xeon(R) Silver 4216	3.0.7-cda111b5812c30d4	N/A	HMAC-SHA2-256
fips.so on Red Hat Enterprise Linux 9 and IBM z16	3.0.7-cda111b5812c30d4	N/A	HMAC-SHA2-256
fips.so on Red Hat Enterprise Linux 9 with PowerVM FW1040.00 with VIOS 3.1.3.00 and IBM POWER10	3.0.7-cda111b5812c30d4	N/A	HMAC-SHA2-256

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

Tested Operational Environments - Software, Firmware, Hybrid:

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Red Hat Enterprise Linux 9	Dell PowerEdge R440	Intel(R) Xeon(R) Silver 4216	Yes	N/A	3.0.7-cda111b5812c30d4
Red Hat Enterprise Linux 9	Dell PowerEdge R440	Intel(R) Xeon(R) Silver 4216	No	N/A	3.0.7-cda111b5812c30d4
Red Hat Enterprise Linux 9	IBM z16 3931-A01	IBM z16	Yes	N/A	3.0.7-cda111b5812c30d4
Red Hat Enterprise Linux 9	IBM z16 3931-A01	IBM z16	No	N/A	3.0.7-cda111b5812c30d4
Red Hat Enterprise Linux 9	IBM 9080-HEX	IBM POWER10	Yes	N/A	3.0.7-cda111b5812c30d4
Red Hat Enterprise Linux 9 with PowerVM FW1040.00 with VIOS 3.1.3.00	IBM 9080-HEX	IBM POWER10	No	N/A	3.0.7-cda111b5812c30d4

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

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

Operating System	Hardware Platform
Red Hat Enterprise Linux 9	Intel(R) Xeon(R) E5

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

There are no components within the cryptographic boundary excluded from the FIPS 140-3 requirements.

2.4 Modes of Operation

Modes List and Description:

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Mode Name	Description	Types of Services	Status Indicator
Approved mode	Automatically entered whenever an approved service is requested	Approved	Equivalent to the indicator of the requested service
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Equivalent to the indicator of the requested service

Table 5: Modes List and Description

After passing all pre-operational self-tests and cryptographic algorithm self-tests executed on start-up, the module automatically transitions to the approved mode. No operator intervention is required to reach this point. The module operates in the approved mode of operation by default and can only transition into the non-approved mode by calling one of the non-approved services listed in the Non-Approved Services table of the Security Policy.

In the operational state, the module accepts service requests from calling applications through its logical interfaces. At any point in the operational state, a calling application can end its process, causing the module to end its operation.

Mode Change Instructions and Status:

The module automatically switches between the approved and non-approved modes depending on the services requested by the operator. The status indicator of the mode of operation is equivalent to the indicator of the service that was requested.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4809, A4810, A4811, A5560, A5576, A5580	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS1	A4809, A4810, A4811, A5560, A5576, A5580	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-65536 Increment 8	SP 800-38A
AES-CBC-CS2	A4809, A4810, A4811, A5560, A5576, A5580	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-65536 Increment 8	SP 800-38A
AES-CBC-CS3	A4809, A4810, A4811, A5560, A5576, A5580	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-65536 Increment 8	SP 800-38A

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
AES-CCM	A4809, A4810, A4811, A5560, A5576, A5580	Key Length - 128, 192, 256 Tag Length - 112, 128, 32, 48, 64, 80, 96 IV Length - IV Length: 56, 64, 72, 80, 88, 96, 104 Payload Length - Payload Length: 256 AAD Length - AAD Length: 0, 256, 65536	SP 800-38C
AES-CFB1	A4809, A4810, A4811, A5560, A5576, A5580	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A4809, A4810, A4811, A5560, A5576, A5580	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A4809, A4810, A4811, A5560, A5576, A5580	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A4809, A4810, A4811, A5560, A5576, A5580	Direction - Generation, Verification Key Length - 128, 192, 256 MAC Length - MAC Length: 128 Message Length - Message Length: 8-524288 Increment 8	SP 800-38B
AES-CTR	A4809, A4810, A4811, A5560, A5576, A5580	Direction - Decrypt, Encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 8-128 Increment 8 Supports Counter larger than maximum value - No Incremental Counter - Yes Counter Tests Performed - Yes	SP 800-38A
AES-ECB	A4809, A4810, A4811, A4837, A4838, A4839, A4840, A4841, A5560, A5576, A5579, A5580, A5586	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A4812, A4815, A4816, A4817, A4818, A4819, A4820, A4821, A4822	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1, 8.2.2 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96, IV Length: 96, 128 Payload Length - Payload Length: 128, 256, 120, 248	SP 800-38D

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
		AAD Length - AAD Length: 128, 256, 120, 0, AAD Length: 64, 96	
AES-GCM	A5577, A5581, A5582, A5583, A5584	Direction - Decrypt, Encrypt IV Generation - External, Internal Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96, IV Length: 96, 128 Payload Length - Payload Length: 128, 256, 120, 248 AAD Length - AAD Length: 128, 256, 120, 0, AAD Length: 64, 96 IV Generation Mode - 8.2.2	SP 800-38D
AES-GMAC	A4812	Direction - Decrypt, Encrypt IV Generation - External IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 AAD Length - AAD Length: 0-65536 Increment 8, AAD Length: 128, 256, 120, 0	SP 800-38D
AES-GMAC	A4815, A4816, A4817, A4818, A4819, A4820, A4821, A4822	Direction - Decrypt, Encrypt IV Generation - External IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 AAD Length - AAD Length: 128, 256, 120, 0	SP 800-38D
AES-GMAC	A5577, A5581, A5582, A5583, A5584	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 AAD Length - AAD Length: 0-65536 Increment 8	SP 800-38D
AES-KW	A4809, A4810, A4811, A5560, A5576, A5580	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 128-4096 Increment 128	SP 800-38F

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
AES-KWP	A4809, A4810, A4811, A5560, A5576, A5580	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 8-4096 Increment 8	SP 800-38F
AES-OFB	A4809, A4810, A4811, A5560, A5576, A5580	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A4809	Direction - Decrypt, Encrypt Key Length - 128, 256 Payload Length - Payload Length: 128-65536 Increment 128, Payload Length: 128-65536 Increment 8 Tweak Mode - Hex Data Unit Length Matches Payload Length - Yes	SP 800-38E
AES-XTS Testing Revision 2.0	A4810, A4811	Direction - Decrypt, Encrypt Key Length - 128, 256 Payload Length - Payload Length: 128-65536 Increment 128 Tweak Mode - Hex Data Unit Length Matches Payload Length - Yes	SP 800-38E
AES-XTS Testing Revision 2.0	A5560, A5576, A5580	Direction - Decrypt, Encrypt Key Length - 128, 256 Payload Length - Payload Length: 128-65536 Increment 8 Tweak Mode - Hex Data Unit Length Matches Payload Length - Yes	SP 800-38E
Counter DRBG	A4808	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Curve - P-224, P-256, P-384, P-521	FIPS 186-5

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigGen (FIPS186-5)	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Component - No	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A4814, A5587	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A4813	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A4814	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A4823, A4824, A4825, A4826	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A5578, A5585	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A5587	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-5
Hash DRBG	A4808	Prediction Resistance - No, Yes Supports Reseed - Yes Mode - SHA-1, SHA2-256, SHA2-512 Entropy Input - Entropy Input: 128, Entropy Input: 256 Nonce - Nonce: 128, Nonce: 64 Personalization String Length - Personalization String Length: 0-512 Increment 128 Additional Input - Additional Input: 0-512 Increment 128 Returned Bits - 160, 256, 512	SP 800-90A Rev. 1

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
HMAC DRBG	A4808	Prediction Resistance - No, Yes Supports Reseed - Yes Mode - SHA-1, SHA2-256, SHA2-512 Entropy Input - Entropy Input: 128, Entropy Input: 256 Nonce - Nonce: 128, Nonce: 64 Personalization String Length - Personalization String Length: 0-512 Increment 128 Additional Input - Additional Input: 0-512 Increment 128 Returned Bits - 160, 256, 512	SP 800-90A Rev. 1
HMAC-SHA-1	A4813, A4823, A4824, A4825, A4826, A5578, A5585	MAC - MAC: 160 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A4813, A4823, A4824, A4825, A4826, A5578, A5585	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A4813, A4823, A4824, A4825, A4826, A5578, A5585	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4813, A4823, A4824, A4825, A4826, A5578, A5585	MAC - MAC: 384 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A4813, A4823, A4824, A4825, A4826, A5578, A5585	MAC - MAC: 512 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A4813, A4823, A4824, A4825, A4826, A5578, A5585	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A4813, A4823, A4824, A4825, A4826, A5578, A5585	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA3-224	A4814, A5587	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A4814, A5587	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A4814, A5587	MAC - MAC: 384 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A4814, A5587	MAC - MAC: 512 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A4845	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
KAS-IFC-SSC	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Modulo - 2048, 3072, 4096, 6144, 8192 Key Generation Methods - rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2-prime-factor Scheme - KAS1 - KAS Role - initiator, responder KAS2 - KAS Role - initiator, responder Fixed Public Exponent - 010001	SP 800-56A Rev. 3
KDA HKDF Sp800-56Cr1	A4807	Fixed Info Pattern - uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-2048 Increment 8 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	SP 800-56C Rev. 2

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
KDA OneStep SP800-56Cr2	A4844	Auxiliary Function Methods - Auxiliary Function Name - SHA-1 Fixed Info Pattern - uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-2048 Increment 8	SP 800-56C Rev. 2
KDF ANS 9.42 (CVL)	A4813, A4823, A4824, A4825, A4826, A5578, A5585	KDF Type - DER Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 zz Length - zz Length: 8-4096 Increment 8 Key Data Length - Key Data Length: 8-4096 Increment 8 Supplemental Information Length - Supplemental Information Length: 0-256 Increment 8 OID - AES-128-KW, AES-192-KW, AES-256-KW	SP 800-135 Rev. 1
KDF ANS 9.42 (CVL)	A4814, A5587	KDF Type - DER Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 zz Length - zz Length: 8-4096 Increment 8 Key Data Length - Key Data Length: 8-4096 Increment 8 Supplemental Information Length - Supplemental Information Length: 0-256 Increment 8 OID - AES-128-KW, AES-192-KW, AES-256-KW	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Field Size - 224, 571 Shared Info Length - Shared Info Length: 0-1024 Increment 8 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4814, A5587	Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 Field Size - 224, 571 Shared Info Length - Shared Info Length: 0-1024 Increment 8 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-135 Rev. 1
KDF SP800-108	A4843	KDF Mode - Counter, Feedback MAC Mode - CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA-1, HMAC-SHA2-224, HMAC-	SP 800-108 Rev. 1

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
		SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/224, HMAC-SHA2-512/256, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 Supported Lengths - Supported Lengths: 8, 72, 128, 776, 3456, 4096 Fixed Data Order - Before Fixed Data Counter Length - 32 Supports Empty IV - Yes Requires Empty IV - No Custom Key In Length - 0	
KDF SSH (CVL)	A4837, A4838, A4839, A4840, A4841, A5579, A5586	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
KTS-IFC	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Function - partialVal IUT ID - 0123456789abcdef Modulo - 2048, 3072, 4096, 6144, 8192 Key Generation Methods - rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2-prime-factor Fixed Public Exponent - 010001 Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Hash Algorithms - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Supports Null Associated Data - Yes Associated Data Pattern - uPartyInfo||vPartyInfo Associated Data Encoding - concatenation Key Length - 768	SP 800-56B Rev. 2
PBKDF	A4813	Iteration Count - Iteration Count: 1000-10000 Increment 1 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Password Length - Password Length: 8-128 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 112-4096 Increment 8, Key Data Length: 128-4096 Increment 8	SP 800-132

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
PBKDF	A4814	Iteration Count - Iteration Count: 1000-10000 Increment 1 HMAC Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 Password Length - Password Length: 8-128 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 112-4096 Increment 8, Key Data Length: 128-4096 Increment 8	SP 800-132
PBKDF	A4823, A4824, A4825, A4826	Iteration Count - Iteration Count: 1000-10000 Increment 1 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Password Length - Password Length: 8-128 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-132
PBKDF	A5578, A5585	Iteration Count - Iteration Count: 1000-10000 Increment 1 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Password Length - Password Length: 8-128 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 112-4096 Increment 8	SP 800-132
PBKDF	A5587	Iteration Count - Iteration Count: 1000-10000 Increment 1 HMAC Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 Password Length - Password Length: 8-128 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 112-4096 Increment 8	SP 800-132
RSA KeyGen (FIPS186-5)	A4813	Key Generation Mode - probableWithProbableAux Modulo - 2048, 3072, 4096, 8192 p mod 8 - 0 Primality Tests - 2powSecStr q mod 8 - 0 Info Generated By Server - No, Yes Private Key Format - standard Public Exponent Mode - random	FIPS 186-5

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
RSA KeyGen (FIPS186-5)	A4823, A4824, A4825, A4826	Key Generation Mode - probableWithProbableAux Modulo - 2048, 3072, 4096, 8192 p mod 8 - 0 Primality Tests - 2powSecStr q mod 8 - 0 Info Generated By Server - No Private Key Format - standard Public Exponent Mode - random	FIPS 186-5
RSA KeyGen (FIPS186-5)	A5578, A5585	Key Generation Mode - probableWithProbableAux Modulo - 2048, 3072, 4096, 8192 p mod 8 - 0 Primality Tests - 2powSecStr q mod 8 - 0 Info Generated By Server - Yes Private Key Format - standard Public Exponent Mode - random	FIPS 186-5
RSA SigGen (FIPS186-5)	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Hash Pair - Hash Algorithm - SHA2-224 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Mask Function - mgf1	FIPS 186-5
RSA SigVer (FIPS186-4)	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024 Hash Pair - Hash Algorithm - SHA2-224 Public Exponent Mode - Random	FIPS 186-4
RSA SigVer (FIPS186-5)	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Hash Pair - Hash Algorithm - SHA3-512 Salt Length - 0 Mask Function - mgf1 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Public Exponent Mode - random	FIPS 186-5
Safe Primes Key Generation	A4845	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3
Safe Primes Key Verification	A4845	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
SHA-1	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA3-224	A4814, A5587	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A4814, A5587	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A4814, A5587	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
SHA3-512	A4814, A5587	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHAKE-128	A4814, A5587	Supports Bit-Oriented Messages - No Supports Empty Message - Yes Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A4814, A5587	Supports Bit-Oriented Messages - No Supports Empty Message - Yes Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	FIPS 202
TLS v1.2 KDF RFC7627 (CVL)	A4813, A4823, A4824, A4825, A4826, A5578, A5585	Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Key Block Length - Key Block Length: 1024	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A4807	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE, PSK, PSK-DHE	SP 800-135 Rev. 1

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
(CKG) Key Pair Generation	Key Type:Asymmetric Key Pair Generation with RSA:2048-16384 bits keys (112-256 bits strength) Key Pair Generation with ECDSA:P-224, P-256, P-384, P-521 curves (112, 128, 192, 256 bits strength) Key Pair Generation with Safe Primes:ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 groups (112-200 bits strength)	N/A	SP 800-133r2 Section 4 example 1

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

The module does not implement non-approved algorithms that are allowed in the approved mode of operation.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

The module does not implement non-approved algorithms that are allowed in the approved mode of operation with no security claimed.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES GCM (external IV)	Encryption
HMAC (< 112-bit keys)	Message authentication
KBKDF, KDA OneStep, HKDF, ANS X9.42 KDF, ANS X9.63 KDF (< 112-bit keys)	KBKDF Key derivation KDA OneStep Key derivation HKDF Key derivation Key derivation with ANS X9.42 KDF Key derivation with ANS X9.63 KDF
KDA OneStep (SHAKE128, SHAKE256)	KDA OneStep Key derivation
ANS X9.42 KDF (SHAKE128, SHAKE256)	Key derivation with ANS X9.42 KDF
ANS X9.63 KDF (SHA-1, SHAKE128, SHAKE256)	Key derivation with ANS X9.63 KDF
SSH KDF (SHA-512/224, SHA-512/256, SHA-3, SHAKE128, SHAKE256)	SSH KDF Key derivation
TLS 1.2 KDF (SHA-1, SHA-224, SHA-512/224, SHA-512/256, SHA-3)	Key derivation with TLS 1.2 KDF
TLS 1.3 KDF (SHA-1, SHA-224, SHA-512, SHA-512/224, SHA-512/256, SHA-3)	Key derivation with TLS 1.3 KDF
PBKDF2 (short password; short salt; insufficient iterations; < 112-bit keys)	Password-based Key derivation
RSA and ECDSA (pre-hashed message)	Signature generation component, Signature verification component
RSA-PSS (invalid salt length)	Signature generation, Signature verification

Table 8: Non-Approved, Not Allowed Algorithms

The table above lists all non-approved cryptographic algorithms of the module employed by the non-approved services of the Non-Approved Services table in Section 4.4.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric Encryption with AES	BC-UnAuth	Symmetric Encryption using AES	Key length:AES-XTS: 128, 256 bits; Other Modes: 128, 192, 256 bits Security strength:AES-XTS: 128, 256 bits; Other Modes: 128, 192, 256 bits	AES-CBC: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CBC-CS1: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CBC-CS2: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CBC-CS3: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CFB1: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CFB128: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CFB8: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CTR: (A4809, A4810, A4811, A5560, A5576, A5580) AES-ECB: (A4809, A4810, A4811, A4837, A4838, A4839, A4840, A4841, A5560, A5576, A5579, A5580, A5586) AES-OFB: (A4809, A4810, A4811, A5560,

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
				A5576, A5580) AES-XTS Testing Revision 2.0: (A4809, A4810, A4811, A5560, A5576, A5580)
Authenticated Symmetric Encryption with AES	BC-Auth	Authenticated Symmetric Encryption using AES	Key length:128, 192, 256 bits Security strength:128, 192, 256 bits	AES-CCM: (A4809, A4810, A4811, A5560, A5576, A5580) AES-KW: (A4809, A4810, A4811, A5560, A5576, A5580) AES-KWP: (A4809, A4810, A4811, A5560, A5576, A5580) AES-GCM: (A4812, A4815, A4816, A4817, A4818, A4819, A4820, A4821, A4822, A5577, A5581, A5582, A5583, A5584)
Symmetric Decryption with AES	BC-UnAuth	Symmetric Decryption using AES	Key length:AES-XTS: 128, 256 bits; Other Modes: 128, 192, 256 bits Security strength:AES-XTS: 128, 256 bits; Other Modes: 128, 192, 256 bits	AES-CBC: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CBC-CS1: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CBC-CS2: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CBC-CS3: (A4809, A4810, A4811, A5560, A5576, A5580)

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
				AES-CFB1: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CFB128: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CFB8: (A4809, A4810, A4811, A5560, A5576, A5580) AES-CTR: (A4809, A4810, A4811, A5560, A5576, A5580) AES-ECB: (A4809, A4810, A4811, A4837, A4838, A4839, A4840, A4841, A5560, A5576, A5579, A5580, A5586) AES-OFB: (A4809, A4810, A4811, A5560, A5576, A5580) AES-XTS Testing Revision 2.0: (A4809, A4810, A4811, A5560, A5576, A5580)
Authenticated Symmetric Decryption with AES	BC-Auth	Authenticated Symmetric Decryption using AES	Key length:128, 192, 256 bits Security strength:128, 192, 256 bits	AES-CCM: (A4809, A4810, A4811, A5560, A5576, A5580) AES-GCM: (A4812, A4815, A4816, A4817, A4818, A4819, A4820, A4821, A4822, A5577,

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
				A5581, A5582, A5583, A5584) AES-KW: (A4809, A4810, A4811, A5560, A5576, A5580) AES-KWP: (A4809, A4810, A4811, A5560, A5576, A5580)
Message Digest	SHA XOF	Compute a message digest	Key length and strength:N/A	SHA-1: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) SHA2-224: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) SHA2-256: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) SHA2-384: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) SHA2-512: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) SHA2-512/224: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) SHA2-512/256: (A4813, A4823, A4824, A4825, A4826, A5578, A5585)

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
				SHA3-224: (A4814, A5587) SHA3-256: (A4814, A5587) SHA3-384: (A4814, A5587) SHA3-512: (A4814, A5587) SHAKE-128: (A4814, A5587) SHAKE-256: (A4814, A5587)
Random Number Generation with CTR_DRBG	DRBG	Generate random bitstrings	Key length:256, 320, 384 bits Security strength:128, 192, 256 bits	Counter DRBG: (A4808)
Random Number Generation with HMAC_DRBG	DRBG	Generate random bitstrings	Key length:320, 512, 1024 bits Security strength:128, 256 bits	HMAC DRBG: (A4808)
Random Number Generation with Hash_DRBG	DRBG	Generate random bitstrings	Key length :880, 1776 bits Security strength:128, 256 bits	Hash DRBG: (A4808)
Message Authentication Code (MAC)	MAC	Compute a MAC tag	HMAC:Hashes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512; Key length: 112-524288 bits; Security strength: 112-256 bits AES CMAC and GMAC:Key length: 128, 192, 256 bits; Security strength: 128, 192, 256 bits	HMAC-SHA-1: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) HMAC-SHA2-224: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) HMAC-SHA2-256: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) HMAC-SHA2-384: (A4813,

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
				A4823, A4824, A4825, A4826, A5578, A5585) HMAC-SHA2-512: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) HMAC-SHA2-512/224: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) HMAC-SHA2-512/256: (A4813, A4823, A4824, A4825, A4826, A5578, A5585) HMAC-SHA3-224: (A4814, A5587) HMAC-SHA3-256: (A4814, A5587) HMAC-SHA3-384: (A4814, A5587) HMAC-SHA3-512: (A4814, A5587) AES-CMAC: (A4809, A4810, A4811, A5560, A5576, A5580) AES-GMAC: (A4812, A4815, A4816, A4817, A4818, A4819, A4820, A4821, A4822, A5577, A5581, A5582, A5583, A5584)

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
Shared Secret Computation with Diffie-Hellman	KAS-SSC	Compute a shared secret using DH	Compliance: SP 800-56A Rev. 3, FIPS 140-3 IG D.F. Scenario 2 (1) Scheme:dpEphem KAS Role:initiator, responder Groups:ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 Security strength:112-200 bits	KAS-FFC-SSC Sp800-56Ar3: (A4845)
Shared Secret Computation with EC Diffie-Hellman	KAS-SSC	Compute a shared secret	Compliance:SP 800-56A Rev. 3, FIPS 140-3 IG D.F. Scenario 2 (1) Scheme:ephemeralUnified KAS Role:initiator, responder Curves:P-224, P-256, P-384, P-521 Security strength:112, 128, 192, 256 bits	KAS-ECC-SSC Sp800-56Ar3: (A4813, A4823, A4824, A4825, A4826, A5578, A5585)
Shared Secret Computation with RSA	KAS-SSC	Compute a shared secret	Key length:2048-15360 bits Security strength:112-256 bits	KAS-IFC-SSC: (A4813, A4823, A4824, A4825, A4826, A5578, A5585)
Key derivation with KBKDF	KBKDF	Perform key derivation	Key length:112-4096 bits Security strength:112-256 bits	KDF SP800-108: (A4843)
Key derivation with KDA OneStep	KAS-56CKDF	Perform key derivation	Key length:224-8192 bits Security strength:112-256 bits	KDA OneStep SP800-56Cr2: (A4844)
Key derivation with HKDF	KAS-56CKDF	Perform key derivation	Key length:224-8192 bits Security strength:112-256 bits	KDA HKDF Sp800-56Cr1: (A4807)
Key derivation with ANS X9.42 KDF	KAS-135KDF	Perform key derivation	Key length:224-8192 bits Security strength:112-256 bits	KDF ANS 9.42: (A4813, A4814, A4823, A4824, A4825, A4826, A5578, A5585, A5587)

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
Key derivation with ANS X9.63 KDF	KAS-135KDF	Perform key derivation	Key length:224-8192 bits Security strength:112-256 bits	KDF ANS 9.63: (A4813, A4814, A4823, A4824, A4825, A4826, A5578, A5585, A5587)
Key derivation with SSH KDF	KAS-135KDF	Perform key derivation	Key length and strength:224-8192 bits Security strength:112-256 bits	KDF SSH: (A4837, A4838, A4839, A4840, A4841, A5579, A5586)
Key derivation with TLS 1.2 KDF	KAS-135KDF	Perform key derivation	Key length:224-8192 bits Security strength:112-256 bits	TLS v1.2 KDF RFC7627: (A4813, A4823, A4824, A4825, A4826, A5578, A5585)
Key derivation with TLS 1.3 KDF	KAS-135KDF	Perform key derivation	Key length:224-8192 bits Security strength:112-256 bits	TLS v1.3 KDF: (A4807)
Password-based Key derivation	PBKDF	Perform key derivation	Compliance:Option 1a from Section 5.4 of SP 800-132 Password length:8-128 characters Password strength:10^8 - 10^128 Salt length:128-4096 bits Iteration count:minimum of 1000 Hashes:SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	PBKDF: (A4813, A4814, A4823, A4824, A4825, A4826, A5578, A5585, A5587)
Digital Signature Generation with RSA	DigSig-SigGen	Generate an RSA signature	Key length:2048-16384 bits Security strength:112-256 bits Padding:PKCS#1 v1.5 and PSS Hashes:SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 IG C.F Compliance:The module supports RSA modulus sizes which	RSA SigGen (FIPS186-5): (A4813, A4823, A4824, A4825, A4826, A5578, A5585)

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
			are not tested by CAVP in compliance with FIPS 140-3 IG C.F	
Digital Signature Verification with RSA	DigSig-SigVer	Verify an RSA signature	RSA SigVer (FIPS 186-5):NIST SP 800-131A Rev. 2 Acceptable; Padding: PKCS#1 v1.5 and PSS; Moduli: 2048-16384 bits; Hashes: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3- 256, SHA3-384, SHA3-512; Security strength: 112-256 bits RSA SigVer (FIPS 186-4):NIST SP 800-131A Rev. 2 Legacy use; Padding: PKCS#1 v1.5 and PSS; Moduli: 1024-2047 bits; Hashes: SHA-224, SHA-256, SHA-384, SHA-512; Security strength: 80-111 bits IG C.F Compliance:The module supports RSA modulus sizes which are not tested by CAVP in compliance with FIPS 140-3 IG C.F	RSA SigVer (FIPS186-4): (A4813, A4823, A4824, A4825, A4826, A5578, A5585) RSA SigVer (FIPS186-5): (A4813, A4823, A4824, A4825, A4826, A5578, A5585)
Digital Signature Generation with ECDSA	DigSig-SigGen	Generate an ECDSA signature	Curves:P-224, P-256, P-384, P-521 Hashes:SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Security strength:112, 128, 192, 256 bits	ECDSA SigGen (FIPS186-5): (A4813, A4814, A4823, A4824, A4825, A4826, A5578, A5585, A5587)
Digital Signature Verification with ECDSA	DigSig-SigVer	Verify an ECDSA signature	Curves:P-224, P-256, P-384, P-521 Hashes:SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Security strength:112, 128, 192, 256 bits	ECDSA SigVer (FIPS186-5): (A4813, A4814, A4823, A4824, A4825, A4826, A5578, A5585, A5587)
Asymmetric Encryption with RSA	AsymKeyPair-Encap	Asymmetric encryption using RSA	Key length:2048-15360 bits Security strength:112-256 bits	KTS-IFC: (A4813, A4823, A4824, A4825,

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
				A4826, A5578, A5585)
Asymmetric Decryption with RSA	AsymKeyPair-Decap	Asymmetric Decryption with RSA	Key length:2048-15360 bits Security strength:112-256 bits	KTS-IFC: (A4813, A4823, A4824, A4825, A4826, A5578, A5585)
Key Pair Generation with RSA	AsymKeyPair-KeyGen CKG	Generate RSA key pair	Key length:2048-16384 bits Security strength:112-256 bits Mode:FIPS 186-5, A.1.6: Probable Primes with Conditions Based on Auxiliary Probable Primes IG C.F Compliance:The module supports RSA modulus sizes which are not tested by CAVP in compliance with FIPS 140-3 IG C.F	RSA KeyGen (FIPS186-5): (A4813, A4823, A4824, A4825, A4826, A5578, A5585) (CKG) Key Pair Generation: ()
Key Pair Generation with ECDSA	AsymKeyPair-KeyGen CKG	Generate ECDSA key pair	Mode:FIPS 186-5 A.2.2: Rejection Sampling Curves:P-224, P-256, P-384, P-521 Security strength:112, 128, 192, 256 bits	ECDSA KeyGen (FIPS186-5): (A4813, A4823, A4824, A4825, A4826, A5578, A5585) (CKG) Key Pair Generation: ()
Key Pair Generation with Safe Primes	AsymKeyPair-KeyGen CKG	Generate DH key pair	Mode:SP 800-56Ar3 Section 5.6.1.1.4 Testing Candidates Groups:MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 Security strength:112-200 bits	Safe Primes Key Generation: (A4845) (CKG) Key Pair Generation: ()
Key Pair Verification with ECDSA	AsymKeyPair-KeyVer	Verify ECDSA key pair	Curves:P-224, P-256, P-384, P-521 Security strength:112, 128, 192, 256 bits	ECDSA KeyVer (FIPS186-5): (A4813, A4823, A4824, A4825, A4826, A5578, A5585)

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
Key Pair Verification with Safe Primes	AsymKeyPair-KeyVer	Verify DH key pair	Mode:SP 800-56Ar3 Sections 5.6.2.1.2 and 5.6.2.1.4 Groups:MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 Security strength:112-200 bits	Safe Primes Key Verification: (A4845)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES GCM IV

For TLS 1.2, the module offers the AES GCM implementation and uses the context of Scenario 1 of FIPS 140-3 IG C.H. The module is compliant with SP 800-52 Rev. 2 Section 3.3.1 and the mechanism for IV generation is compliant with RFC 5288 and 8446.

The module does not implement the TLS protocol. The module's implementation of AES GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS protocol implicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key.

In the event the module's power is lost and restored, the consuming application must ensure that a new key for use with the AES GCM key encryption or decryption under this scenario shall be established.

Alternatively, the Crypto Officer can use the module's API to perform AES GCM encryption using internal IV generation. These IVs are always 96 bits and generated using the approved DRBG internal to the module's boundary, compliant to Scenario 2 of FIPS 140-3 IG C.H.

The module also provides a non-approved AES GCM encryption service which accepts arbitrary external IVs from the operator. This service can be requested by invoking the EVP_EncryptInit_ex2 API function with a non-NULL IV value. When this is the case, the API will set a non-approved service indicator.

Finally, for TLS 1.3, the AES GCM implementation uses the context of Scenario 5 of FIPS 140- 3 IG C.H. The protocol that provides this compliance is TLS 1.3, defined in RFC8446 of August 2018, using the cipher-suites that explicitly select AES GCM as the encryption/decryption cipher (Appendix B.4 of RFC8446). The module supports acceptable AES GCM cipher suites from Section 3.3.1 of SP 800-52 Rev. 2. The module's implementation of AES GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS protocol implicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key.

2.7.2 AES XTS

The length of a single data unit encrypted or decrypted with AES XTS shall not exceed 2^{20} AES blocks, that is 16MB, of data per XTS instance. An XTS instance is defined in Section 4 of SP 800-38E.

To meet the requirement stated in IG C.I, the module implements a check that ensures, before performing any cryptographic operation, that the two AES keys used in AES XTS mode are not identical. Key_1 and Key_2 shall

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

be generated and/or established independently according to the rules for component symmetric keys from NIST SP 800-133rev2, Sec. 6.3.

The XTS mode shall only be used for the cryptographic protection of data on storage devices. It shall not be used for other purposes, such as the encryption of data in transit.

2.7.3 Key Derivation using SP 800-132 PBKDF2

The module provides password-based key derivation (PBKDF2), compliant with SP 800-132. The module supports option 1a from Section 5.4 of SP 800-132, in which the Master Key (MK) or a segment of it is used directly as the Data Protection Key (DPK). In accordance to SP 800-132 and FIPS 140-3 IG D.N, the following requirements are met:

- Derived keys shall be used only for storage applications, and shall not be used for any other purposes. The length of the MK or DPK is 112 bits or more.
- Passwords or passphrases, used as an input for the PBKDF2, shall not be used as cryptographic keys.
- The minimum length of the password or passphrase accepted by the module is 8 characters. The probability of guessing the value, assuming a worst-case scenario of all digits, is estimated to be at most 10-8. Combined with the minimum iteration count as described below, this provides an acceptable trade-off between user experience and security against brute-force attacks.
- A portion of the salt shall be generated randomly using the SP 800-90A Rev. 1 DRBG provided by the module. The minimum length required is 128 bits.
- The iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The minimum value accepted by the module is 1000.

If any of these requirements are not met, the requested service is non-approved (see Non-Approved Services table in Section 4.4).

2.7.4 SP 800-56A Rev. 3 Assurances

To comply with the assurances found in Section 5.6.2 of SP 800-56A Rev. 3, the operator must use the module together with an application that implements the TLS protocol. Additionally, the module's approved key pair generation service (see Approved Services table in Section 4.3) must be used to generate ephemeral Diffie-Hellman or EC Diffie-Hellman key pairs, or the key pairs must be obtained from another FIPS-validated module. As part of this service, the module will internally perform the full public key validation of the generated public key.

The module's shared secret computation service will internally perform the full public key validation of the peer public key, complying with Sections 5.6.2.2.1 and 5.6.2.2.2 of SP 800-56A Rev. 3.

2.7.5 SHA-3

The module implements the SHA-3 algorithms as both standalone and part of higher-level algorithms (in compliance with FIPS 140-3 IG C.C). As detailed in Section 2.6 with corresponding certificates, the cryptographic algorithms that use of SHA-3 include RSA signature generation and verification, ECDSA signature generation and verification, KBKDF, KDA HKDF, X9.63 KDF, X9.42 KDF, PBKDF, OneStep KDA, and HMAC. In addition, the implementation of the extendable output functions SHAKE128 and SHAKE256 were verified to have a standalone usage.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.7.6 RSA Signatures

The module implements only the approved modulus sizes of 2048, 3072, and 4096 bits for signature generation.

For signature verification, the module implements the approved module sizes of 1024, 2048, 3072, and 4096 bits. Each algorithm was tested, and corresponding certificates can be found detailed in Section 2.6.

The module also supports RSA signature verification with 1280, 1536 and 1792 modulus bits. These modulus sizes are allowed only for legacy use, in compliance with FIPS 140-3 IG C.F.

2.7.7 Key Agreement

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

2.7.8 Key Transport

The module does not establish SSPs using an approved key transport scheme (KTS). However, it does offer approved authenticated algorithms that can be used by an external operator/application as part of an approved KTS.

2.8 RBG and Entropy

Cert Number	Vendor Name
E48	Red Hat, LLC

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
RHEL OpenSSL FIPS provider CPU Time Jitter Entropy source	Non-Physical	Red Hat Enterprise Linux 9 on Dell PowerEdge R440 on Intel® Xeon® Silver 4216; Red Hat Enterprise Linux 9 on IBM z16 3931-A01 on IBM z16; Red Hat Enterprise Linux 9 on PowerVM FW1040.00 with VIOS 3.1.3.00 on IBM 9080 HEX on IBM POWER10	256 bits	224 bits	LFSR, HMAC-SHA-512 DRBG, AES-256 CTR DRBG

Table 11: Entropy Sources

The module employs two Deterministic Random Bit Generator (DRBG) implementations based on SP 800-90A Rev. 1. These DRBGs are used internally by the module (e.g. to generate seeds for asymmetric key pairs and random numbers for security functions). They can also be accessed using the specified API functions. The following parameters are used:

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

1. Private DRBG: AES-256 CTR_DRBG with derivation function. This DRBG is used to generate secret random values (e.g. during asymmetric key pair generation). It can be accessed using RAND_priv_bytes.
2. Public DRBG: AES-256 CTR_DRBG with derivation function. This DRBG is used to generate general purpose random values that do not need to remain secret (e.g. initialization vectors). It can be accessed using RAND_bytes.

These DRBGs will always employ prediction resistance. More information regarding the configuration and design of these DRBGs can be found in the module's manual pages.

2.9 Key Generation

The module implements Cryptographic Key Generation (CKG, vendor affirmed), compliant with SP 800-133 Rev. 2. When random values are required, they are obtained from the SP 800-90A Rev. 1 approved DRBG, compliant with Section 4 of SP 800-133 Rev. 2 (without XOR):

- Safe primes key pair generation: compliant with SP 800-133 Rev. 2, Section 5.2, which maps to SP 800-56A Rev. 3. The method described in Section 5.6.1.1.4 of SP 800-56A Rev. 3 ("Testing Candidates") is used.
- RSA key pair generation: compliant with SP 800-133 Rev. 2, Section 5.1, which maps to FIPS 186-4. The method described in Appendix B.3.6 of FIPS 186-4 ("Probable Primes with Conditions Based on Auxiliary Probable Primes") is used.
- ECC (ECDH and ECDSA) key pair generation: compliant with SP 800-133 Rev. 2, Section 5.1, which maps to FIPS 186-4. The method described in Appendix B.4.2 of FIPS 186-4 ("Testing Candidates") is used.

Additionally, the module implements the following key derivation methods:

- KBKDF: compliant with SP 800-108 Rev. 1. This implementation can be used to generate secret keys from a pre-existing key-derivation-key.
- KDA OneStep, HKDF: compliant with SP 800-56C Rev. 2. These implementations shall only be used to generate secret keys in the context of an SP 800-56A Rev. 3 key agreement scheme.
- ANS X9.42 KDF, ANS X9.63 KDF: compliant with SP 800-135 Rev. 1. These implementations shall only be used to generate secret keys in the context of an ANS X9.42-2001 resp. ANS X9.63-2001 key agreement scheme.
- SSH KDF, TLS 1.2 KDF, TLS 1.3 KDF: compliant with SP 800-135 Rev. 1. These implementations shall only be used to generate secret keys in the context of the SSH, TLS 1.2, or TLS 1.3 protocols, respectively.
- PBKDF2: compliant with option 1a of SP 800-132. This implementation shall only be used to derive keys for use in storage applications.

Intermediate key generation values are not output from the module and are explicitly zeroized after processing the service.

2.10 Key Establishment

The module provides Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) shared secret computation compliant with SP800-56A Rev. 3, in accordance with scenario 2 (1) of FIPS 140-3 IG D.F.

For Diffie-Hellman, the module supports the use of the safe primes defined in RFC 3526 (IKE) and RFC 7919 (TLS). Note that the module only implements key pair generation, key pair verification, and shared secret

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

computation. No other part of the IKE or TLS protocols is implemented (with the exception of the TLS 1.2 and 1.3 KDFs):

- IKE (RFC 3526):
  - MODP-2048 (ID = 14)
  - MODP-3072 (ID = 15)
  - MODP-4096 (ID = 16)
  - MODP-6144 (ID = 17)
  - MODP-8192 (ID = 18)
- TLS (RFC 7919)
  - ffdhe2048 (ID = 256)
  - ffdhe3072 (ID = 257)
  - ffdhe4096 (ID = 258)
  - ffdhe6144 (ID = 259)
  - ffdhe8192 (ID = 260)

For Elliptic Curve Diffie-Hellman, the module supports the NIST-defined P-224, P-256, P-384, and P-521 curves.

According to FIPS 140-3 IG D.B, the key sizes of DH and ECDH provide the following security strengths in the approved mode of operation:

- Diffie-Hellman shared secret computation provides between 112 and 200 bits of security strength.
- EC Diffie-Hellman shared secret computation provides between 112 and 256 bits of security strength.

In addition, the module provides RSA shared secret computation compliant with SP800-56B Rev. 2, in accordance with scenario 1 (1) of FIPS 140-3 IG D.F.

For RSA key generation, the module provides 2048-15360 bits keys. Therefore, according to FIPS 140-3 IG D.B, the key sizes of RSA shared secret computation provide 112-256 bits of security strength in the approved mode of operation.

The module offers RSA key wrapping and unwrapping using KTS-OAEP-basic scheme. The implementation supports 2048-15360 bits modulus size, with both key encapsulation and un-encapsulation supported. The module does not implement key confirmation. The SSP establishment methodology provides 112 to 256 bits of encryption strength.

The module also supports the AES KW, AES KWP, and AES GCM key wrapping mechanisms. These algorithms can be used to wrap SSPs with a security strength between 128 and 256 bits, depending on the wrapping key size.

2.11 Industry Protocols

The module implements the SSH key derivation function for use in the SSH protocol (RFC 4253 and RFC 6668).

GCM with internal IV generation in the approved mode is compliant with versions 1.2 and 1.3 of the TLS protocol (RFC 5288 and 8446) and shall only be used in conjunction with the TLS protocol. Additionally, the module implements the TLS 1.2 and TLS 1.3 key derivation functions for use in the TLS protocol.

No parts of the SSH, TLS, or IKE protocols, other than those mentioned above, have been tested by the CAVP and CMVP.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API input parameters
N/A	Data Output	API output parameters
N/A	Control Input	API function calls
N/A	Status Output	API return codes, error queue

Table 12: Ports and Interfaces

The logical interfaces are the APIs through which the applications request services. These logical interfaces are logically separated from each other by the API design. The module does not implement a control output interface.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module does not support authentication methods.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 13: Roles

The module supports the Crypto Officer role only. This sole role is implicitly and always assumed by the operator of the module when performing a service. The module does not support multiple concurrent operators.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
HKDF Key derivation	Perform key derivation	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	HKDF Derived Key	Key derivation with HKDF	Crypto Officer - Shared secret: W,E - HKDF Derived Key: G,R
Key derivation with TLS 1.3 KDF	Perform key derivation	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	TLS 1.3 Derived Key	Key derivation with TLS 1.3 KDF	Crypto Officer - Shared secret: W,E - TLS 1.3 KDF Derived key: G,R

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Random Number Generation	Generate random bitstrings	EVP_RAND_generate returns 1	Output length	Random bytes	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG	Crypto Officer - Entropy input: W,E - DRBG seed: E,W,G - Internal state (V, C): G,W,E - Internal state (V, Key): W,E,G
Symmetric Encryption	Used to perform symmetric encryption of an entry plaintext	EVP_EncryptFinal_ex returns 1	Plaintext, AES Key	Ciphertext	Symmetric Encryption with AES	Crypto Officer - AES Key: W,E
Symmetric Decryption	Used to perform symmetric decryption of an entry ciphertext	EVP_DecryptFinal_ex returns 1	Ciphertext, AES Key	Plaintext	Symmetric Decryption with AES	Crypto Officer - AES Key: W,E

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message Authentication Code (MAC)	Compute a MAC tag	HMAC: OSSL_MAC_PARAM_REDHAT_FIPS_INDICATOR_APPROVED; Others: EVP_MAC_final returns 1	Message, AES key or HMAC key	MAC tag	Message Authentication Code (MAC)	Crypto Officer - HMAC key: W,E - AES Key: W,E
Authenticated Symmetric Encryption	Authenticated Symmetric Encryption with AES	AES GCM: EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED; Others: EVP_EncryptFinal_ex returns 1	Plaintext, AES key, IV	Ciphertext, MAC tag	Authenticated Symmetric Encryption with AES	Crypto Officer - AES Key: W,E
Authenticated Symmetric Decryption	Authenticated Symmetric Decryption with AES	AES GCM: EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED; Others: EVP_DecryptFinal_ex returns 1	Ciphertext, AES key, IV, MAC tag	Plaintext or failure	Authenticated Symmetric Decryption with AES	Crypto Officer - AES Key: W,E
Key Pair Generation	Generate key pair	EVP_PKEY_generate returns 1	Group or Curve or Modulus bits	DH key pair; EC key pair; RSA key pair	Key Pair Generation with RSA Key Pair Generation with ECDSA Key Pair Generation with Safe Primes	Crypto Officer - Module - generated RSA Private key: G,R - Module - generated RSA Public key:

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,E - Module - generated EC Private key: G,R - Module - generated EC Public key: G,E - Module - generated DH Private key: G,R - Module - generated DH Public key: G,E - Intermediate key generation value: G,E,Z

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Key Pair Verification	Verify ECDSA or Safe Primes key pair	EVP_PKEY_public_check or EVP_PKEY_private_check or EVP_PKEY_check returns 1	Key pair	Pass/fail	Key Pair Verification with ECDSA Key Pair Verification with Safe Primes	Crypto Officer - EC private key: W,E - EC public key: W,E - DH private key: W,E - DH public key: W,E
Digital Signature Generation	Generate a digital signature	RSA: OSSL_RH_FIPSINDICATOR_APPROVED and EVP_SIGNATURE_REDHAT_FIPS_INDICATOR_APPROVED; ECDSA: OSSL_RH_FIPSINDICATOR_APPROVED	Message, Private key	Signature	Digital Signature Generation with RSA Digital Signature Generation with ECDSA	Crypto Officer - RSA private key: W,E - EC private key: W,E
Digital Signature Verification	Verify a digital signature	RSA: OSSL_RH_FIPSINDICATOR_APPROVED and EVP_SIGNATURE_REDHAT_FIPS_INDICATOR_APPROVED; ECDSA: OSSL_RH_FIPSINDICATOR_APPROVED	Message, Public key, signature	Pass/fail	Digital Signature Verification with RSA Digital Signature Verificat	Crypto Officer - RSA public key: W,E - EC public key: W,E

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					ion with ECDSA	
Shared Secret Computation	Compute a shared secret	EVP_PKEY_derive returns 1	Private key, Public key (peer)	Shared Secret	Shared Secret Computation with RSA Shared Secret Computation with Diffie-Hellman Shared Secret Computation with EC Diffie-Hellman	Crypto Officer - RSA private key: W,E - RSA public key: W,E - EC private key: W,E - EC public key: W,E - DH private key: W,E - DH public key: W,E - Shared Secret: G,R
Key derivation with ANS X9.42 KDF	Perform key derivation	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	ANS X9.42 Derived Key	Key derivation with ANS X9.42 KDF	Crypto Officer - Shared secret: W,E - ANS X9.42 KDF

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Derived key: G,R
Key derivation with ANS X9.63 KDF	Perform key derivation	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	ANS X9.63 Derived Key	Key derivation with ANS X9.63 KDF	Crypto Officer - Shared secret: W,E - ANS X9.63 KDF Derived key: G,R
Asymmetric Encryption	Perform RSA-based encryption (compliant with SP 800-56B Rev. 2))	EVP_PKEY_REDHAT_FIPS_INDICATOR_APPROVED	RSA public key, plaintext key	Encapsulated key	Asymmetric Encryption with RSA	Crypto Officer - RSA public key: W,E
Asymmetric Decryption	Perform RSA-based decryption (compliant with SP 800-56B Rev. 2))	EVP_PKEY_REDHAT_FIPS_INDICATOR_APPROVED	RSA private key, Encapsulated key	plaintext key	Asymmetric Decryption with RSA	Crypto Officer - RSA private key: W,E
Password-based Key	Perform key derivation	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Password or passphrase	PBKDF Derived Key	Password-based Key	Crypto Officer - Passwo

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
derivation					derivation	rd: W,E - PBKDF2 Derived key: G,R
Message digest	Used to generate a SHA-1, SHA-2, or SHA-3/SHAKE message digest	EVP_DigestFinal_ex returns 1	Message	Message digest	Message Digest	Crypto Officer
Key derivation with TLS 1.2 KDF	Perform key derivation	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	TLS 1.2 Derived Key	Key derivation with TLS 1.2 KDF	Crypto Officer - Shared secret: W,E - TLS 1.2 KDF Derived key: G,R
SSH KDF Key derivation	Perform key derivation	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	SSH Derived Key	Key derivation with SSH KDF	Crypto Officer - Shared secret: W,E - SSH KDF Derived key: G,R

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
KBKDF Key derivation	Perform key derivation	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Key-derivation key	KBKDF Derived Key	Key derivation with KBKDF	Crypto Officer - Key-derivation key: W,E - KBKDF Derived Key: G,R
KDA OneStep Key derivation	Perform key derivation	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	KDA OneStep Derived Key	Key derivation with KDA OneStep	Crypto Officer - Shared secret: W,E - KDA OneStep Derived Key: G,R
Show status	Show the current status of the module	None	N/A	Module status	None	Crypto Officer
Show module name and version	Show module name and the version of the module	None	N/A	Name and version information	None	Crypto Officer
Self-test	Perform CASTs and	None	N/A	Pass/fail result	Message Digest Message	Crypto Officer

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	integrity test			of self-tests	Authentication Code (MAC) Symmetric Encryption with AES Symmetric Decryption with AES Authenticated Symmetric Encryption with AES Authenticated Symmetric Decryption with AES Digital Signature Generation with RSA Digital Signature Generation with ECDSA Digital Signature	

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Verification with RSA Digital Signature Verification with ECDSA Key derivation with KBKDF Key derivation with KDA OneStep Key derivation with HKDF Key derivation with ANS X9.42 KDF Key derivation with ANS X9.63 KDF Key derivation with SSH KDF Key derivation with TLS 1.2 KDF	

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Key derivation with TLS 1.3 KDF Password-based Key derivation Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG Shared Secret Computation with Diffie-Hellman Shared Secret Computation with EC Diffie-Hellman	

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Zeroization	Zeroize SSPs	None	Any SSP	N/A	None	Crypto Officer - AES Key: Z - HMAC key: Z - RSA private key: Z - RSA public key: Z - DH private key: Z - DH public key: Z - EC private key: Z - EC public key: Z - Key-derivation key: Z - Password: Z - PBKDF2 Derived key: Z - KBKDF Derived Key: Z - HKDF Derived

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key: Z - SSH KDF Derived key: Z - ANS X9.42 KDF Derived key: Z - ANS X9.63 KDF Derived key: Z - KDA OneStep Derived Key: Z - TLS 1.2 KDF Derived key: Z - TLS 1.3 KDF Derived key: Z - Shared Secret: Z - Entropy input: Z - DRBG seed: Z - Internal state

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						(V, C): Z - Internal state (V, Key): Z - Intermediate key generation value: Z

Table 14: Approved Services

The module provides services to operators that assume the available role. All services are described in detail in the API documentation (manual pages). The convention below applies when specifying the access permissions (types) that the service has for each SSP:

- **Generate (G):** The module generates or derives the SSP.
- **Read (R):** The SSP is read from the module (e.g. the SSP is output).
- **Write (W):** The SSP is updated, imported, or written to the module.
- **Execute (E):** The module uses the SSP in performing a cryptographic operation.
- **Zeroize (Z):** The module zeroizes the SSP.

To interact with the module, a calling application must use the EVP API layer provided by OpenSSL. This layer will delegate the request to the FIPS provider, which will in turn perform the requested service. Additionally, this EVP API layer can be used to retrieve the approved service indicator for the module. The redhat_ossl_query_fipsindicator() function indicates whether an EVP API function is approved.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
AES GCM (external IV)	Authenticated Encryption	AES GCM (external IV)	CO
HMAC (< 112-bit keys)	Message authentication	HMAC (< 112-bit keys)	CO
Key derivation	Derive a key from a key-derivation key or a shared secret	KBKDF, KDA OneStep, HKDF, ANS X9.42 KDF, ANS X9.63 KDF (< 112-bit keys)	CO

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Algorithms	Role
		KDA OneStep (SHAKE128, SHAKE256) ANS X9.42 KDF (SHAKE128, SHAKE256) ANS X9.63 KDF (SHA-1, SHAKE128, SHAKE256) SSH KDF (SHA-512/224, SHA-512/256, SHA-3, SHAKE128, SHAKE256) TLS 1.2 KDF (SHA-1, SHA-224, SHA-512/224, SHA-512/256, SHA-3) TLS 1.3 KDF (SHA-1, SHA-224, SHA-512, SHA-512/224, SHA-512/256, SHA-3)	
PBKDF2	Password-based Key derivation	PBKDF2 (short password; short salt; insufficient iterations; < 112-bit keys)	CO
Signature generation	Generate a signature	RSA and ECDSA (pre-hashed message) RSA-PSS (invalid salt length)	CO
Signature verification	Verify a signature	RSA and ECDSA (pre-hashed message) RSA-PSS (invalid salt length)	CO

Table 15: Non-Approved Services

The table above lists the non-approved services in this module, the algorithms involved and the roles that can request the service. In this table, CO specifies the Crypto Officer role.

4.5 External Software/Firmware Loaded

The module does not load external software or firmware.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified by comparing a HMAC SHA-256 value calculated at run time with the HMAC SHA-256 value embedded in the fips.so file that was computed at build time. This operation is performed by the verify_integrity() function which performs a KAT for the HMAC SHA-256 algorithm in order to test its proper operation before performing the checksum of the module file.

5.2 Initiate on Demand

Integrity tests are performed as part of the pre-operational self-tests, which are executed when the module is initialized. The integrity test may be invoked on-demand by unloading and subsequently re-initializing the module, or by calling the OSSL_PROVIDER_self_test function. This will perform (among others) the software integrity test.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

Any SSPs contained within the module are protected by the process isolation and memory separation mechanisms, and only the module has control over these SSPs.

6.2 Configuration Settings and Restrictions

The module shall be installed as stated in Section 11. If properly installed, the operating system provides process isolation and memory protection mechanisms that ensure appropriate separation for memory access among the processes on the system. Each process has control over its own data and uncontrolled access to the data of other processes is prevented.

Instrumentation tools like the ptrace system call, gdb and strace, userspace live patching, as well as other tracing mechanisms offered by the Linux environment such as ftrace or systemtap, shall not be used in the operational environment. The use of any of these tools implies that the cryptographic module is running in a non-validated operational environment.

6.3 Additional Information

The Red Hat Enterprise Linux operating system is used as the basis of other products which include but are not limited to:

- Red Hat Enterprise Linux CoreOS
- Red Hat Ansible Automation Platform
- Red Hat OpenStack Platform
- Red Hat OpenShift
- Red Hat Gluster Storage
- Red Hat Satellite

Compliance is maintained for these products whenever the binary is found unchanged.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

7 Physical Security

The module is comprised of software only and therefore this section is not applicable.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

8 Non-Invasive Security

This module does not implement any non-invasive security mechanism and therefore this section is not applicable.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	Temporary storage for SSPs used by the module as part of service execution. The module does not perform persistent storage of SSPs	Dynamic

Table 16: Storage Areas

The module does not perform persistent storage of SSPs. The SSPs are temporarily stored in the RAM in plaintext form. SSPs are provided to the module by the calling process and are destroyed when released by the appropriate zeroization function calls.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API input parameters	Operator calling application (TOEPP)	Cryptographic module	Plaintext	Manual	Electronic	
API output parameters	Cryptographic module	Operator calling application (TOEPP)	Plaintext	Manual	Electronic	

Table 17: SSP Input-Output Methods

The module only supports SSP entry and output to and from the calling application running on the same operational environment. This corresponds to manual distribution, electronic entry/output (“CM Software to/from App via TOEPP Path”) per FIPS 140-3 IG 9.5.A Table 1.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Wipe and Free memory block allocated	Zeroizes the SSPs contained within the cipher handle.	By calling the appropriate zeroization functions: AES key: EVP_CIPHER_CTX_free and EVP_MAC_CTX_free; HMAC key: EVP_MAC_CTX_free; Key-derivation key: EVP_KDF_CTX_free; Shared Secret: EVP_KDF_CTX_free; Password: EVP_KDF_CTX_free; KBKDF Derived Key: EVP_KDF_CTX_free; HKDF Derived Key: EVP_KDF_CTX_free; TLS Derived Key: EVP_KDF_CTX_free; SSH KDF Derived key:	By calling the cipher related zeroization API

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Zeroization Method	Description	Rationale	Operator Initiation
		EVP_KDF_CTX_free; ANS X9.63 KDF Derived key: EVP_KDF_CTX_free; ANS X9.42 KDF Derived key: EVP_KDF_CTX_free; PBKDF2 Derived Key: EVP_KDF_CTX_free; KDA OneStep Derived Key: EVP_KDF_CTX_free; Entropy input: EVP_RAND_CTX_free; Internal state (V, Key), Internal state (V, C): EVP_RAND_CTX_free; DH public & private key: EVP_PKEY_free; EC public & private key: EVP_PKEY_free; RSA public & private key: EVP_PKEY_free	
Automatic	Automatically zeroized by the module when no longer needed	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable.	N/A
Module Reset	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed.	By unloading and reloading the module

Table 18: SSP Zeroization Methods

The application that uses the module is responsible for the appropriate zeroization of SSPs. The module provides key allocation and destruction functions, which overwrites the memory occupied by the SSP´s information with zeros before its deallocation.

Memory allocation of SSPs is performed by the OPENSSL_malloc() API call and the application in use of the module is responsible for the calling of the appropriate zeroization functions from the OpenSSL API. The zeroization functions then overwrite the memory occupied by SSPs and de-allocate the memory with the OPENSSL_free() call. OPENSSL_cleanse() should be used to overwrite sensitive data such as private keys.

In case of abnormal termination, or swap in/out of a physical memory page of a process, the SSPs in physical memory are overwritten by the Linux kernel before the physical memory is allocated to another process.

All data output is inhibited during zeroization.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES Key	AES Key used for Encryption Decryption; Message authentication ;	AES-XTS: 128, 256 bits Rest of modes: 128, 192, 256 bits - AES-XTS:	Symmetric key - CSP			Symmetric Encryption with AES Authenticated Symmetric Encryption

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		128, 256 bits Rest of modes: 128, 192, 256 bits				with AES Symmetric Decryption with AES Authenticated Symmetric Decryption with AES Message Authentication Code (MAC)
HMAC key	HMAC key used for Message authentication;	112-524288 bits - 112-256 bits	Symmetric key - CSP			Message Authentication Code (MAC)
Key-derivation key	Key-derivation key used for KBKDF Key derivation;	112-4096 bits - 112-256 bits	Symmetric key - CSP			Key derivation with KBKDF
Shared secret	Shared secret generated by ECDH/DH/RSA shared secret computation	224-8912 bits - 112-256 bits	Shared Secret - CSP		Shared Secret Computation with Diffie-Hellman Shared Secret Computation with EC Diffie-Hellman Shared Secret Computation with RSA	Key derivation with HKDF Key derivation with TLS 1.3 KDF Key derivation with ANS X9.42 KDF Key derivation with ANS X9.63 KDF Key derivation with TLS 1.2 KDF

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						Key derivation with SSH KDF Key derivation with KDA OneStep Shared Secret Computation with Diffie-Hellman Shared Secret Computation with EC Diffie-Hellman Shared Secret Computation with RSA
Password	Password used for Password-based Key derivation	8-128 characters - 10^8 - 10^128	Password - CSP			Password-based Key derivation
KBKDF Derived Key	KBKDF Derived Key used for KBKDF Key derivation	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation with KBKDF		Key derivation with KBKDF
KDA OneStep Derived Key	KDA OneStep Derived Key used for KDA OneStep Key derivation	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key derivation with KDA OneStep		Key derivation with KDA OneStep
HKDF Derived Key	HKDF Derived Key used for HKDF Key derivation;	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key derivation with HKDF		Key derivation with HKDF

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ANS X9.42 KDF Derived key	ANS X9.42 KDF Derived key used for Key derivation with ANS X9.42 KDF;	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key derivation with ANS X9.42 KDF		Key derivation with ANS X9.42 KDF
ANS X9.63 KDF Derived key	ANS X9.63 KDF Derived key used for Key derivation with ANS X9.63 KDF;	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key derivation with ANS X9.63 KDF		Key derivation with ANS X9.63 KDF
SSH KDF Derived key	SSH KDF Derived key used for SSH KDF Key derivation;	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key derivation with SSH KDF		Key derivation with SSH KDF
TLS 1.2 KDF Derived key	TLS 1.2 KDF Derived key used for Key derivation with TLS 1.2 KDF;	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key derivation with TLS 1.2 KDF		
TLS 1.3 KDF Derived key	TLS 1.3 KDF Derived key used for Key derivation with TLS 1.2 KDF;	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key derivation with TLS 1.3 KDF		
PBKDF2 Derived key	PBKDF2 Derived key used for Password-based Key derivation;	224-8192 bits - 112-256 bits	Symmetric key - CSP	Password-based Key derivation		Password-based Key derivation
Entropy input	Entropy input used for Random	128-384 bits - 112-336 bits	Entropy Input - CSP			Random Number Generation

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	number generation					with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG
DRBG seed	DRBG seed used for Random number generation (IG D.L compliant)	CTR_DRBG: 256, 320, 384 bits; Hash_DRBG: 440, 888 bits; HMAC_DRBG: 440 bits, 888 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG, Hash_DRBG: 128, 256 bits	Seed - CSP	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG		
Internal state (V, Key)	Internal state (V, Key) used for Random number generation (IG D.L compliant)	CTR_DRBG: 256, 320, 384 bits, HMAC_DRBG: 320, 512, 1024 bits - CTR_DRBG: 128, 192, 256 bits, HMAC_DRBG: 128, 256 bits	Internal state - CSP	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG		

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Internal state (V, C)	Internal state (V, C) used for Random number generation (IG D.L compliant)	880, 1776 bits - 128, 256 bits	Internal state - CSP	Random Number Generation with Hash_DRBG		
DH private key	DH private key written to the module	2048-8192 bits - 112-200 bits	Private key - CSP			Shared Secret Computation with Diffie-Hellman Key Pair Verification with Safe Primes
DH public key	DH Public key written to the module	2048-8192 bits - 112-200 bits	Public key - PSP			Key Pair Verification with Safe Primes Shared Secret Computation with Diffie-Hellman
Module-generated DH Private key	DH Private key generated by the module	2048-8192 bits - 112-200 bits	Private key - CSP	Key Pair Generation with Safe Primes		
Module-generated DH Public key	DH Public key generated by the module	2048-8192 bits - 112-200 bits	Public key - PSP	Key Pair Generation with Safe Primes		
EC private key	EC private key written the module and used by ECDSA and ECDH	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Private key - CSP			Shared Secret Computation with EC Diffie-Hellman Digital Signature Generation

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						with ECDSA Key Pair Verification with ECDSA
EC public key	EC Public key written the module and used by ECDSA and ECDH	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Public key - PSP			Digital Signature Verification with ECDSA Shared Secret Computation with EC Diffie-Hellman Key Pair Verification with ECDSA
Module-generated EC Private key	EC Private key generated by the module	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Private key - CSP	Key Pair Generation with ECDSA		
Module-generated EC Public key	EC Public key generated by the module	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Public key - PSP	Key Pair Generation with ECDSA		
RSA private key	RSA private key written to the module	2048-15360 bits - 112-256 bits	Private key - CSP			Shared Secret Computation with RSA Asymmetric Decryption with RSA Digital Signature Generation with RSA
RSA public key	RSA public key written to the module	Signature verification: 1024-15360	Public key - PSP			Shared Secret Computation with RSA

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		bits; Others: 2048-15360 bits - Signature verification: 80-256 bits; Others: 112-256 bits				Asymmetric Encryption with RSA Digital Signature Verification with RSA
Module-generated RSA Private key	RSA private key generated by the module	2048-15360 bits - 112-256 bits	Private key - CSP	Key Pair Generation with RSA		
Module-generated RSA Public key	RSA public key generated by the module	2048-15360 bits - Key pair generation: 112-256 bits	Public key - PSP	Key Pair Generation with RSA		
Shared Secret	Shared secret generated by ECDH/DH/RSA shared secret computation	224-8912 bits - 112-256 bits	Shared Secret - CSP		Shared Secret Computation with Diffie-Hellman Shared Secret Computation with EC Diffie-Hellman Shared Secret Computation with RSA	Key derivation with KDA OneStep Key derivation with ANS X9.42 KDF Key derivation with ANS X9.63 KDF Key derivation with SSH KDF Key derivation with HKDF Shared Secret Computation with Diffie-Hellman Shared Secret Computation

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						with EC Diffie-Hellman Shared Secret Computation with RSA
Intermediate key generation value	Intermediate key pair generation value generated during key generation and key derivation services (SP 800-133 Rev. 2 Section 4, 5.1, and 5.2)	112-15360 bits - 112-256 bits	Intermediate value - CSP	Key Pair Generation with ECDSA Key Pair Generation with RSA Key Pair Generation with Safe Primes		Key Pair Generation with ECDSA Key Pair Generation with RSA Key Pair Generation with Safe Primes

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	
HMAC key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	
Key-derivation key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	KBKDF Derived Key:Derives

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Shared secret	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	KDA OneStep Derived Key:Derives HKDF Derived Key:Derives ANS X9.42 KDF Derived key:Derives ANS X9.63 KDF Derived key:Derives SSH KDF Derived key:Derives TLS 1.2 KDF Derived key:Derives TLS 1.3 KDF Derived key:Derives RSA private key:Established By RSA public key:Established By EC private key:Established By EC public key:Established By DH private key:Established By DH public key:Established By
Password	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	PBKDF2 Derived key:Derives
KBKDF Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Key-derivation key:Derived From
KDA OneStep Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block	Shared secret:Derived From

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				allocated Module Reset	
HKDF Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Shared secret:Derived From
ANS X9.42 KDF Derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Shared secret:Derived From
ANS X9.63 KDF Derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Shared secret:Derived From
SSH KDF Derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Shared secret:Derived From
TLS 1.2 KDF Derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Shared secret:Derived From
TLS 1.3 KDF Derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Shared secret:Derived From
PBKDF2 Derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Password:Derived From

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Entropy input		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Automatic Module Reset	DRBG seed:Derives
DRBG seed	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Automatic Module Reset	Entropy input:Derived From Internal state (V, Key):Generates Internal state (V, C):Generates
Internal state (V, Key)	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Automatic Module Reset	DRBG seed:Generated From
Internal state (V, C)	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Automatic Module Reset	DRBG seed:Generated From
DH private key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH public key:Paired With
DH public key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH private key:Paired With
Module-generated DH Private key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block	Module-generated DH Public key:Paired With Intermediate key

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				allocated Module Reset	generation value:Generated From
Module-generated DH Public key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	Module-generated DH Private key:Paired With Intermediate key generation value:Generated From
EC private key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	EC public key:Paired With
EC public key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	EC private key:Paired With
Module-generated EC Private key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	Module-generated EC Public key:Paired With Intermediate key generation value:Generated From
Module-generated EC Public key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	Module-generated EC private key:Paired With Intermediate key generation value:Generated From
RSA private key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block	RSA public key:Paired With

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				allocated Module Reset	
RSA public key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	RSA private key:Paired With
Module-generated RSA Private key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	Module-generated RSA public key:Paired With Intermediate key generation value:Generated From
Module-generated RSA Public key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	Module-generated RSA private key:Paired With Intermediate key generation value:Generated From
Shared Secret	API input parameters API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	DH private key:Established By DH public key:Established By EC private key:Established By EC public key:Established By HKDF Derived Key:Derives KDA OneStep Derived Key:Derives TLS 1.2 KDF Derived key:Derives TLS 1.3 KDF Derived key:Derives SSH KDF Derived key:Derives ANS X9.63 KDF

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					Derived key:Derives ANS X9.42 KDF Derived key:Derives RSA private key:Established By RSA public key:Established By
Intermediate key generation value	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Automatic	DH private key:Generates DH public key:Generates EC private key:Generates EC public key:Generates RSA private key:Generates RSA public key:Generates

Table 20: SSP Table 2

9.5 Transitions

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes except signature verification, starting January 1, 2031.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256	256-bit key	Message authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test of the fips.so shared library component of the module. Verified by comparing an HMAC SHA-256 value calculated at run time with the HMAC SHA-256 value embedded in the fips.so file that was computed at build time.

Table 21: Pre-Operational Self-Tests

The pre-operational software integrity tests are performed automatically when the module is initialized, before the module transitions into the operational state. While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed. The module transitions to the operational state only after the pre-operational self-tests are passed successfully.

Prior the first use, a CAST is executed for the algorithms used in the Pre-operational Self-Tests.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC	SHA2-256	KAT	CAST	Module becomes operational	Encryption	At power-on, before the integrity test
AES-GCM - Encryption	Key size: 256 bits	KAT	CAST	Module becomes operational	Encryption	At power-on, before the integrity test
AES-GCM - Decryption	Key size: 256 bits	KAT	CAST	Module becomes operational	Decryption	At power-on, before the integrity test
AES-ECB	Key size: 128 bits	KAT	CAST	Module becomes operational	Decryption	At power-on, before the integrity test

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1	24-bit message	KAT	CAST	Module becomes operational	Message digest	At power-on, before the integrity test
SHA2-512	24-bit message	KAT	CAST	Module becomes operational	Message digest	At power-on, before the integrity test
SHA3-256	32-bit message	KAT	CAST	Module becomes operational	Message digest	At power-on, before the integrity test
KBKDF	HMAC-SHA2-256 in counter mode	KAT	CAST	Module becomes operational	Key derivation	At power-on, before the integrity test
KDA OneStep	SHA2-224	KAT	CAST	Module becomes operational	Key derivation	At power-on, before the integrity test
KDA HKDF	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	At power-on, before the integrity test
X9.42 KDF	SHA-1	KAT	CAST	Module becomes operational	Key derivation	At power-on, before the integrity test
X9.63 KDF	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	At power-on, before the integrity test
SSH KDF	SHA-1	KAT	CAST	Module becomes operational	Key derivation	At power-on, before the integrity test
TLS 1.2 KDF	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	At power-on, before the integrity test
TLS 1.3 KDF	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	At power-on, before the integrity test

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
PBKDF2	SHA-256 with 4096 iterations, 24-byte password, and 288-bit salt	KAT	CAST	Module becomes operational	Key derivation	At power-on, before the integrity test
Counter DRBG	AES-128 with prediction resistance, with derivation function	KAT	CAST	Module becomes operational	Instantiate, seed, generate, reseed, generate (compliant to SP 800-90Ar1 Section 11.3)	At power-on, before the integrity test
Hash DRBG	SHA2-256 with prediction resistance	KAT	CAST	Module becomes operational	Instantiate, seed, generate, reseed, generate (compliant to SP 800-90Ar1 Section 11.3)	At power-on, before the integrity test
HMAC DRBG	HMAC-SHA-1 with prediction resistance	KAT	CAST	Module becomes operational	Instantiate, seed, generate, reseed, generate (compliant to SP 800-90Ar1 Section 11.3)	At power-on, before the integrity test
KAS-FFC-SSC	ffdhe2048	KAT	CAST	Module becomes operational	Shared secret computation	At power-on, before the integrity test
KAS-ECC-SSC	P-256	KAT	CAST	Module becomes operational	Shared secret computation	At power-on, before the integrity test
ECDSA SigGen (FIPS186-5)	SHA-256 and P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Signature generation	At power-on, before the integrity test
ECDSA SigVer (FIPS186-5)	SHA-256 and P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Signature verification	At power-on, before the integrity test
RSA PKCS#1 v1.5 SigGen	SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Signature generation	At power-on, before the integrity test

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA PKCS#1 v1.5 SigVer	SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Signature verification	At power-on, before the integrity test
DH	N/A	PCT	PCT	Key pair generation is successful	SP 800-56Ar3 Section 5.6.2.1.4	Key pair generation
ECDSA KeyGen (FIPS186-5)	SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation
RSA PKCS#1 v1.5 KeyGen	SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation

Table 22: Conditional Self-Tests

Data output through the data output interface is inhibited during the conditional self-tests. The module does not return control to the calling application until the tests are completed. If any of these tests fails, the module transitions to the error state (Section 10.4).

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256	Message authentication	SW/FW Integrity	On demand	Manually

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC	KAT	CAST	On demand	Manually
AES-GCM - Encryption	KAT	CAST	On demand	Manually
AES-GCM - Decryption	KAT	CAST	On demand	Manually
AES-ECB	KAT	CAST	On demand	Manually
SHA-1	KAT	CAST	On demand	Manually
SHA2-512	KAT	CAST	On demand	Manually
SHA3-256	KAT	CAST	On demand	Manually

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KBKDF	KAT	CAST	On demand	Manually
KDA OneStep	KAT	CAST	On demand	Manually
KDA HKDF	KAT	CAST	On demand	Manually
X9.42 KDF	KAT	CAST	On demand	Manually
X9.63 KDF	KAT	CAST	On demand	Manually
SSH KDF	KAT	CAST	On demand	Manually
TLS 1.2 KDF	KAT	CAST	On demand	Manually
TLS 1.3 KDF	KAT	CAST	On demand	Manually
PBKDF2	KAT	CAST	On demand	Manually
Counter DRBG	KAT	CAST	On demand	Manually
Hash DRBG	KAT	CAST	On demand	Manually
HMAC DRBG	KAT	CAST	On demand	Manually
KAS-FFC-SSC	KAT	CAST	On demand	Manually
KAS-ECC-SSC	KAT	CAST	On demand	Manually
ECDSA SigGen (FIPS186-5)	KAT	CAST	On demand	Manually
ECDSA SigVer (FIPS186-5)	KAT	CAST	On demand	Manually
RSA PKCS#1 v1.5 SigGen	KAT	CAST	On demand	Manually
RSA PKCS#1 v1.5 SigVer	KAT	CAST	On demand	Manually
DH	PCT	PCT	On demand	Manually
ECDSA KeyGen (FIPS186-5)	PCT	PCT	On demand	Manually
RSA PKCS#1 v1.5 KeyGen	PCT	PCT	On demand	Manually

Table 24: Conditional Periodic Information

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	If the module fails any of the self-tests, the module enters the error state. In the error state, the module immediately stops functioning and ends the application process	Software integrity test failure CAST failure	Module reinitialization	OSSL_PROV_PARAM_STATUS is set to 0. Module will not load.
PCT Error	Pairwise consistency test failure	PCT failure	Module reinitialization	Module is aborted

Table 25: Error States

If the module fails any of the self-tests, the module enters the error state. In the error state, the module immediately stops functioning and ends the application process. Consequently, the data output interface is inhibited, and the module no longer accepts inputs or requests (as the module is no longer running).

10.5 Operator Initiation of Self-Tests

Both conditional and pre-operational self-tests can be executed on-demand by unloading and subsequently re-initializing the module, or by calling the OSSL_PROVIDER_self_test function. The pair-wise consistency tests can be invoked on demand by requesting the key pair generation service.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is distributed as a part of the Red Hat Enterprise Linux 9 (RHEL 9) by means of one the following RPM packages:

- openssl-fips-provider-3.0.7-11.el9_0
- openssl-fips-provider-3.0.7-11.el9_8

Before the openssl-fips-provider-3.0.7-11.el9_0 RPM package or the openssl-fips-provider-3.0.7-11.el9_8 RPM package is installed, the RHEL 9 system must operate in the approved mode. This can be achieved by:

- Starting the installation in the approved mode. Add the fips=1 option to the kernel command line during the system installation. During the software selection stage, do not install any third-party software.
- Switching the system into the approved mode after the installation. Execute the fips-mode-setup –enable command. Restart the system. More information can be found from the vendor documentation.

In both cases, the Crypto Officer must verify the RHEL 9 system operates in the approved mode by executing the fips-mode-setup --check command, which should output "FIPS mode is enabled."

11.2 Administrator Guidance

After installation of openssl-fips-provider-3.0.7-11.el9_0 RPM package or of openssl-fips-provider-3.0.7-11.el9_8 RPM package, the Crypto Officer must execute the openssl list -providers command. The Crypto Officer must ensure that the fips provider is listed in the output as follows:

```
fips
  name: Red Hat Enterprise Linux 9 - OpenSSL FIPS Provider
  version: 3.0.7-cda111b5812c30d4
  status: active
```

The cryptographic boundary consists only of the FIPS provider as listed. If any other OpenSSL or third-party provider is invoked, the user is not interacting with the module specified in this Security Policy.

11.3 Non-Administrator Guidance

There is no non-administrator guidance.

11.4 Design and Rules

Not applicable for this module.

11.5 Maintenance Requirements

There are no maintenance requirements.

11.6 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of unloading the module. This will zeroize all SSPs in volatile memory. Then, if desired, the installed RPM package (openssl-fips-provider-3.0.7-11.el9_0 or openssl-fips-provider-3.0.7-11.el9_8) can be uninstalled from the RHEL 9 system.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

12 Mitigation of Other Attacks

Certain cryptographic subroutines and algorithms are vulnerable to timing analysis. The module mitigates this vulnerability by using constant-time implementations. This includes, but is not limited to:

- Big number operations: computing GCDs, modular inversion, multiplication, division, and modular exponentiation (using Montgomery multiplication)
- Elliptic curve point arithmetic: addition and multiplication (using the Montgomery ladder)
- Vector-based AES implementations

In addition, RSA, ECDSA, ECDH, and DH employ blinding techniques to further impede timing and power analysis. No configuration is needed to enable the aforementioned countermeasures.

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
API	Application Programming Interface
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter
CTS	Ciphertext Stealing
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EVP	Envelope
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
GMAC	Galois Counter Mode Message Authentication Code
HKDF	HMAC-based Key Derivation Function
HMAC	Keyed-Hash Message Authentication Code
IKE	Internet Key Exchange
KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-based Key Derivation Function

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

KTS	Key Transport Scheme
KW	Key Wrap
KWP	Key Wrap with Padding
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OAEP	Optimal Asymmetric Encryption Padding
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PCT	Pair-wise Consistency Test
PBKDF2	Password-based Key Derivation Function v2
PKCS	Public-Key Cryptography Standards
PSS	Probabilistic Signature Scheme
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSH	Secure Shell
SSP	Sensitive Security Parameter
TLS	Transport Layer Security
XOF	Extendable Output Function
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Appendix B. References

ANS X9.42-2001	**Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography** 2001 https://webstore.ansi.org/standards/ascx9/ansix9422001
ANS X9.63-2001	**Public Key Cryptography for the Financial Services Industry, Key Agreement and Key Transport Using Elliptic Curve Cryptography** 2001 https://webstore.ansi.org/standards/ascx9/ansix9632001
FIPS 140-3	**FIPS PUB 140-3 - Security Requirements For Cryptographic Modules** March 2019 https://doi.org/10.6028/NIST.FIPS.140-3
FIPS 140-3 IG	**Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program** 2 September 2025 https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
FIPS 180-4	**Secure Hash Standard (SHS)** March 2012 https://doi.org/10.6028/NIST.FIPS.180-4
FIPS 186-4	**Digital Signature Standard (DSS)** July 2013 https://doi.org/10.6028/NIST.FIPS.186-4
FIPS 186-5	**Digital Signature Standard (DSS)** February 2023 https://doi.org/10.6028/NIST.FIPS.186-5
FIPS 197	**Advanced Encryption Standard** May 2023 https://doi.org/10.6028/NIST.FIPS.197-upd1
FIPS 198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 https://doi.org/10.6028/NIST.FIPS.198-1

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

FIPS 202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 https://doi.org/10.6028/NIST.FIPS.202
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1** February 2003 https://www.ietf.org/rfc/rfc3447.txt
RFC 3526	**More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)** May 2003 https://www.ietf.org/rfc/rfc3526.txt
RFC 5288	**AES Galois Counter Mode (GCM) Cipher Suites for TLS** August 2008 https://www.ietf.org/rfc/rfc5288.txt
RFC 7919	**Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS)** August 2016 https://www.ietf.org/rfc/rfc7919.txt
RFC 8446	**The Transport Layer Security (TLS) Protocol Version 1.3** August 2018 https://www.ietf.org/rfc/rfc8446.txt
SP 800-38A	**Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 https://doi.org/10.6028/NIST.SP.800-38A
SP 800-38A Addendum	**Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode** October 2010 https://doi.org/10.6028/NIST.SP.800-38A-Add
SP 800-38B	**Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication** May 2005 https://doi.org/10.6028/NIST.SP.800-38B
SP 800-38C	**Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality** May 2004 https://doi.org/10.6028/NIST.SP.800-38C

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

SP 800-38D	**Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 https://doi.org/10.6028/NIST.SP.800-38D
SP 800-38E	**Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices** January 2010 https://doi.org/10.6028/NIST.SP.800-38E
SP 800-38F	**Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 https://doi.org/10.6028/NIST.SP.800-38F
SP 800-52 Rev. 2	**Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations** August 2019 https://doi.org/10.6028/NIST.SP.800-52r2
SP 800-56A Rev. 3	**Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography** April 2018 https://doi.org/10.6028/NIST.SP.800-56Ar3
SP 800-56B Rev. 2	**Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography** March 2019 https://doi.org/10.6028/NIST.SP.800-56Br2
SP 800-56C Rev. 2	**Recommendation for Key-Derivation Methods in Key-Establishment Schemes** August 2020 https://doi.org/10.6028/NIST.SP.800-56Cr2
SP 800-90A Rev. 1	**Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 https://doi.org/10.6028/NIST.SP.800-90Ar1
SP 800-90B	**Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://doi.org/10.6028/NIST.SP.800-90B

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

SP 800-108 Rev. 1	**NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions** August 2022 https://doi.org/10.6028/NIST.SP.800-108r1
SP 800-131A Rev. 2	**Transitioning the Use of Cryptographic Algorithms and Key Lengths** March 2019 https://doi.org/10.6028/NIST.SP.800-131Ar2
SP 800-132	**Recommendation for Password-Based Key Derivation - Part 1: Storage Applications** December 2010 https://doi.org/10.6028/NIST.SP.800-132
SP 800-133 Rev. 2	**Recommendation for Cryptographic Key Generation** June 2020 https://doi.org/10.6028/NIST.SP.800-133r2
SP 800-135 Rev. 1	**Recommendation for Existing Application-Specific Key Derivation Functions** December 2011 https://doi.org/10.6028/NIST.SP.800-135r1

© 2026 Red Hat, LLC/ atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.