

SUSE, LLC

SUSE Linux Enterprise Libica Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Version 1.2

Last update: 2025-11-07

Prepared by:

atsec information security corporation

4516 Seton Center Parkway, Suite 250

Austin, TX 78759

www.atsec.com

© 2025 SUSE, 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
1.2 Security Levels 6
2 Cryptographic Module Specification 7
2.1 Description 7
2.2 Tested and Vendor Affirmed Module Version and Identification 8
2.3 Excluded Components 9
2.4 Modes of Operation 9
2.5 Algorithms 10
2.6 Security Function Implementations 14
2.7 Algorithm Specific Information 17
2.7.1 AES XTS 17
2.7.2 AES GCM IV 17
2.7.3 Key Agreement 17
2.7.4 SHA-1 17
2.8 RBG and Entropy 18
2.9 Key Generation 18
2.10 Key Establishment 19
2.11 Industry Protocols 19
3 Cryptographic Module Interfaces 20
3.1 Ports and Interfaces 20
4 Roles, Services, and Authentication 21
4.1 Authentication Methods 21
4.2 Roles 21
4.3 Approved Services 21
4.4 Non-Approved Services 27
4.5 External Software/Firmware Loaded 27
5 Software/Firmware Security 28
5.1 Integrity Techniques 28
5.2 Initiate on Demand 28
6 Operational Environment 29
6.1 Operational Environment Type and Requirements 29

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

6.2 Configuration Settings and Restrictions 29
7 Physical Security 30
7.1 Mechanisms and Actions Required 30
8 Non-Invasive Security 32
9 Sensitive Security Parameters Management 33
9.1 Storage Areas 33
9.2 SSP Input-Output Methods 33
9.3 SSP Zeroization Methods 33
9.4 SSPs 34
9.5 Transitions 38
10 Self-Tests 39
10.1 Pre-Operational Self-Tests 39
10.2 Conditional Self-Tests 39
10.3 Periodic Self-Test Information 44
10.4 Error States 46
10.5 Operator Initiation of Self-Tests 47
11 Life-Cycle Assurance 48
11.1 Installation, Initialization, and Startup Procedures 48
11.1.1 Module Installation 48
11.1.2 Operating Environment Configuration 48
11.1.3 Module Installation for Vendor Affirmed Platforms 49
11.2 Administrator Guidance 49
11.3 Non-Administrator Guidance 49
11.4 End of Life 50
12 Mitigation of Other Attacks 51
Appendix A. Glossary and Abbreviations 52
Appendix B. References 54

© 2025 SUSE, 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 6
Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets) 8
Table 3: Tested Module Identification – Hybrid Disjoint Hardware 8
Table 4: Tested Operational Environments - Software, Firmware, Hybrid 9
Table 5: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid 9
Table 6: Modes List and Description 10
Table 7: Approved Algorithms 13
Table 8: Vendor-Affirmed Algorithms 14
Table 9: Non-Approved, Not Allowed Algorithms 14
Table 10: Security Function Implementations 17
Table 11: Entropy Certificates 18
Table 12: Entropy Sources 18
Table 13: Ports and Interfaces 20
Table 14: Roles 21
Table 15: Approved Services 27
Table 16: Non-Approved Services 27
Table 17: Storage Areas 33
Table 18: SSP Input-Output Methods 33
Table 19: SSP Zeroization Methods 34
Table 20: SSP Table 1 36
Table 21: SSP Table 2 38
Table 22: Pre-Operational Self-Tests 39
Table 23: Conditional Self-Tests 44
Table 24: Pre-Operational Periodic Information 44
Table 25: Conditional Periodic Information 46
Table 26: Error States 46
Table 27: Installation for Vendor Affirmed Platforms 49
Table 28: RPM packages 49

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

List of Figures

Figure 1 – Cryptographic Boundary 8
Figure 2 – IBM z15 Mainframe Computer 30
Figure 3 – IBM z15 Processor Unit Chip 31

© 2025 SUSE, 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 software version 1.1 and hardware version "IBM z/15" of the SUSE Linux Enterprise Libica Cryptographic Module. It has a one-to-one mapping to SP 800-140B starting with Section B.2.1 named “General” that maps to Section 1 in this document and ending with Section B.2.12 named “Mitigation of other attacks” that maps to Section 12 in this document.

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	1
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

© 2025 SUSE, 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 SUSE Linux Enterprise Libica Cryptographic Module (hereafter referred to as "the module") is a software-hybrid module that provides general purpose cryptographic algorithms to applications running in the user space of the underlying operating system through a C language application program interface (API).

The module is composed of a software library, which provides the API and a subset of the cryptographic algorithms, and the Central Processor Assist for Cryptographic Functions (CPACF), which is part of the z15 processor and provides cryptographic algorithms implemented in hardware. In addition, the module uses the SUSE Linux Enterprise OpenSSL Cryptographic Module as a bound module (also referred to as "the bound OpenSSL module"), which provides additional algorithms not implemented in Libica. The SUSE Linux Enterprise OpenSSL Cryptographic Module has been submitted under INTU for full FIPS 140-3 validation with TID-11-2191.

For the purpose of the FIPS 140-3 validation, the module is a software-hybrid, multi-chip standalone cryptographic module validated at overall security level 1.

Module Type: Software-hybrid

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The software block diagram below shows the cryptographic boundary of the module, and its interfaces with the operational environment.

Tested Operational Environment's Physical Perimeter (TOEPP):

The tested operational environment's physical is the general-purpose computer on which the software component of the module runs. This operational environment includes the hardware component of the module. A physical entropy source is present within the operational environment, but is outside of the cryptographic boundary.

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

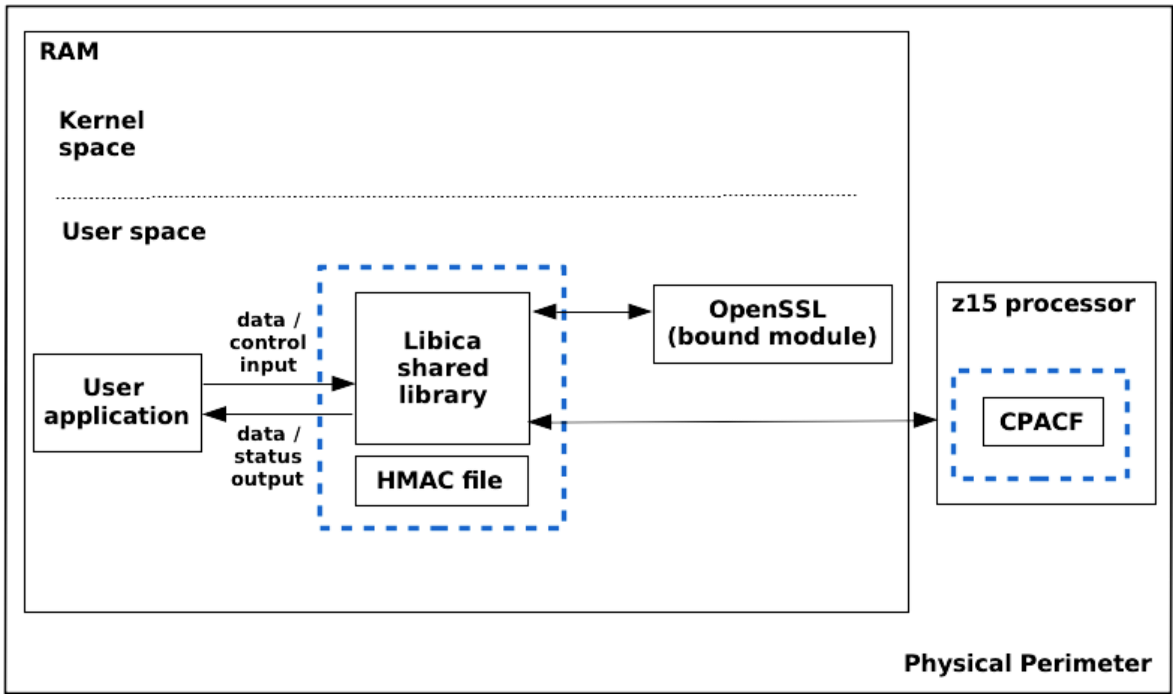


Figure 1 – Cryptographic Boundary

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
libica.so.4.2.1 and .libica.so.4.2.1.hmac on SUSE Linux Enterprise Server 15 SP4 and IBM z15	1.1	N/A	HMAC-SHA-256

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

Tested Module Identification – Hybrid Disjoint Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
IBM z/15 with FC3863	IBM z15	N/A	z15	N/A

Table 3: Tested Module Identification – Hybrid Disjoint Hardware

The IBM z/15 hardware must have Feature Code 3863 (FC3863) enabled. This feature code enables processor acceleration implementations (PAI) for AES and ECC algorithms.

Tested Operational Environments - Software, Firmware, Hybrid:

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

The module has been tested on the following platforms with the corresponding module variants and configuration options:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
SUSE Linux Enterprise Server 15 SP4	IBM z/15 with FC3863	z15	Yes	N/A	1.1

Table 4: Tested Operational Environments - Software, Firmware, Hybrid

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

In addition to the platforms listed above, SUSE has also tested the module on the platforms in the following table, and claims vendor affirmation on them.

Note: the 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.

Operating System	Hardware Platform
SUSE Linux Enterprise Server 15 SP4	IBM LinuxONE III LT1 with FC3863 [z15]
SUSE Linux Enterprise Micro 5.3	IBM z/15 with FC3863 [z15]
SUSE Linux Enterprise Micro 5.3	IBM LinuxONE III LT1 with FC3863 [z15]
SUSE Linux Enterprise Base Container Image 15SP4	IBM z/15 with FC3863 [z15]
SUSE Linux Enterprise Base Container Image 15SP4	IBM LinuxONE III LT1 with FC3863 [z15]

Table 5: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

2.3 Excluded Components

The module does not contain any excluded components.

2.4 Modes of Operation

Modes List and Description:

The module supports the modes of operation shown in the following table:

© 2025 SUSE, 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 (ica_get_fip s_indicator () returns .fips_approved = 1)
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Equivalent to the indicator of the requested service (ica_get_fip s_indicator () does not return .fips_approved = 1)

Table 6: Modes List and Description

When the module starts up successfully, after passing all the pre-operational self-tests, the module is operating in the approved mode of operation by default and can only be transitioned into the non-Approved mode by calling one of the non-Approved services listed in Section 4.4. Please see Section 4.3 for the details on service indicator provided by the module that identifies when an approved service is called.

Critical security parameters (CSPs) used or stored in the approved mode are not used in non-approved mode, and vice versa. The module creates separate contexts for each cryptographic service; therefore, the CSPs used in approved services and non-approved services are separated by design and not shared.

2.5 Algorithms

Approved Algorithms:

The following table lists all approved algorithms of the Libica module and corresponding CAVP certificates. Approved algorithms with CAVP certificates A3147 and A3150 are provided by the bound OpenSSL module.

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3378	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS1	A3378	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 256-65536 Increment 8	SP 800-38A
AES-CBC-CS2	A3378	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 256-65536 Increment 8	SP 800-38A
AES-CBC-CS3	A3378	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 256-65536 Increment 8	SP 800-38A
AES-CCM	A3378	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	SP 800-38C

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
		Payload Length - Payload Length: 8-256 Increment 8 AAD Length - AAD Length: 8, 256, 65536	
AES-CFB128	A3378	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A3378	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3378	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	A3378	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	A3378	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A3377, A3378	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.2 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 Payload Length - Payload Length: 128-65536 Increment 128 AAD Length - AAD Length: 128, 256, 120, 0	SP 800-38D
AES-GMAC	A3377, A3378	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.2 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-OFB	A3378	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

© 2025 SUSE, 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-XTS Testing Revision 2.0	A3378	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
Counter DRBG	A3150	Prediction Resistance - No, Yes Supports Reseed - No Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - No, Yes Additional Input - Additional Input: 0 Entropy Input - Entropy Input: 128, Entropy Input: 192, Entropy Input: 256, Entropy Input: 320, Entropy Input: 384 Nonce - Nonce: 128, Nonce: 64, Nonce: 96 Personalization String Length - Personalization String Length: 0 Returned Bits - 1024, 512	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-4)	A3147	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3378	Component - Yes Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3378	Component - Yes Curve - P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
HMAC-SHA2-256	A3147	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A3378	Domain Parameter Generation Methods - P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
RSA KeyGen (FIPS186-4)	A3147	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2	FIPS 186-4

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	CAVP Cert	Properties	Reference
		Info Generated By Server - No Public Exponent Mode - Random Private Key Format - Standard	
SHA-1	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-224	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-256	A3147	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-384	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512/224	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512/256	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA3-224	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 202
SHA3-256	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 202
SHA3-384	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 202
SHA3-512	A3378	Message Length - Message Length: 8-65536 Increment 8	FIPS 202
SHAKE-128	A3378	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	A3378	Supports Bit-Oriented Messages - No Supports Empty Message - Yes Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	FIPS 202

Table 7: Approved Algorithms

Vendor-Affirmed Algorithms:

The following table lists the vendor-affirmed algorithms provided by the bound OpenSSL module.

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Properties	Implementation	Reference
CKG		N/A	SP 800-133 Rev. 2 Section 4 Example 1 with V=0

Table 8: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

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:

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:

The table below lists non-approved algorithms implemented in the Libica module that are not allowed in the approved mode of operation. These algorithms are used by the non-approved services listed in Section 4.4. The Libica module does not use any non-approved algorithm implemented in the bound OpenSSL module.

Name	Use and Function
RSA modulus exponentiation primitive.	Sign, verify, encrypt, and decrypt primitives.
DRBG	Random Number Generation service

Table 9: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Key pair generation	AsymKeyPair-KeyGen CKG	Key pair generation	Provided by:Bound OpenSSL module	ECDSA KeyGen (FIPS186-4): (A3147) RSA KeyGen (FIPS186-4): (A3147) CKG: ()
Message authentication code (MAC) with HMAC	MAC	Integrity test	Provided by:Bound OpenSSL module	HMAC-SHA2-256: (A3147) SHA2-256: (A3147)
Random number generation	DRBG	Random number generation	Provided by:Bound OpenSSL module	Counter DRBG: (A3150)

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
Authenticated symmetric encryption	BC-Auth	Authenticated encryption		AES-GCM: (A3377, A3378) AES-CCM: (A3378)
Authenticated symmetric decryption	BC-Auth	Authenticated decryption		AES-GCM: (A3377, A3378) AES-CCM: (A3378)
Message authentication code (MAC) with AES	MAC	Message authentication code (MAC)		AES-GMAC: (A3377, A3378) AES-CMAC: (A3378)
Symmetric encryption	BC-UnAuth	Symmetric encryption		AES-CBC: (A3378) AES-CBC-CS1: (A3378) AES-CBC-CS2: (A3378) AES-CBC-CS3: (A3378) AES-CFB128: (A3378) AES-CFB8: (A3378) AES-CTR: (A3378) AES-ECB: (A3378) AES-OFB: (A3378) AES-XTS Testing Revision 2.0: (A3378)
Symmetric decryption	BC-UnAuth	Symmetric decryption		AES-CBC: (A3378) AES-CBC-CS1: (A3378) AES-CBC-CS2: (A3378) AES-CBC-CS3: (A3378) AES-CFB128:

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
				(A3378) AES-CFB8: (A3378) AES-CTR: (A3378) AES-ECB: (A3378) AES-OFB: (A3378) AES-XTS Testing Revision 2.0: (A3378)
Digital signature generation	DigSig-SigGen	Digital signature generation		ECDSA SigGen (FIPS186-4): (A3378)
Digital signature verification	DigSig-SigVer	Digital signature verification (usage of SHA-1 is considered Legacy Use)		ECDSA SigVer (FIPS186-4): (A3378)
Shared secret computation	KAS-SSC	Shared secret computation		KAS-ECC-SSC Sp800-56Ar3: (A3378)
Message digest	SHA XOF	Message digest		SHA-1: (A3378) SHA2-224: (A3378) SHA2-256: (A3378) SHA2-384: (A3378) SHA2-512: (A3378) SHA2-512/224: (A3378) SHA2-512/256: (A3378) SHA3-224: (A3378) SHA3-256: (A3378) SHA3-384: (A3378) SHA3-512:

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Type	Description	Properties	Algorithms
				(A3378) SHAKE-128: (A3378) SHAKE-256: (A3378)

Table 10: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES XTS

The AES algorithm in XTS mode can be only used for the cryptographic protection of data on storage devices, as specified in SP 800-38E. The length of a single data unit encrypted with the XTS-AES shall not exceed 2^{20} AES blocks, that is, 16MB of data.

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. As the module does not implement symmetric key generation, this check is performed when the keys are input by the operator. AES-XTS keys (i.e., Key_1 and Key_2) entered into the module shall be generated and/or established independently according to NIST SP 800-133rev2, Section 6.3. for an approved use of AES-XTS.

2.7.2 AES GCM IV

The AES GCM IV generation is in compliance with section 8.2.2 of SP 800-38D and IG C.H scenario 2 of the FIPS 140-3 IG, in which the GCM IV is generated internally and entirety randomly. The DRBG provided by the OpenSSL bound module, compliant with SP 800-90A Rev. 1, is used for generating the IV. The DRBG is fully seeded with entropy provided by the non-physical entropy source that is not within the cryptographic boundary of the module but is within the tested operational environment's physical perimeter.

The GCM IV must be at least 96 bits in length, which is enforced by the module.

When a GCM IV is used for decryption, the responsibility for the IV generation lies with the party that performs the AES GCM encryption.

2.7.3 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.

2.7.4 SHA-1

The SHA-1 algorithm is offered "as is" in approved mode: Usage of SHA-1 is only approved when used for Message Digest and Message Authentication HMAC, and Signature Verification (Legacy). The module doesn't offer signature generation with SHA-1 in approved mode. The SHA-1 algorithm as implemented by the module will be non-approved for all purposes, starting January 1, 2031.

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.8 RBG and Entropy

The Deterministic Random Bit Generator (DRBG) is provided by the bound OpenSSL module. The DRBG is based on SP 800-90A Rev. 1 for the creation of RSA and ECDSA keys in OpenSSL, and for the random generation of the IV used in AES GCM and the k value used in ECDSA signature generation. The use of the DRBG provided by the bound OpenSSL module is only internal. Notice that the Libica module implements a separate DRBG for providing a random generation number service, but this algorithm implementation, as well as the service, are considered non-approved.

The DRBG is initialized during initialization of the bounded OpenSSL module; this module loads the DRBG by default using the CTR_DRBG mechanism with AES-256, with derivation function, and without prediction resistance.

The bound OpenSSL module uses an SP 800-90B compliant entropy source specified in the below tables. This entropy source is located within the physical perimeter, but outside of the cryptographic boundary of the module. The module obtains 384 bits to seed the DRBG, and 256 bits to reseed it, sufficient to provide a DRBG with 256 bits of security strength.

Cert Number	Vendor Name
E28	SUSE LLC
E29	SUSE LLC

Table 11: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
CPU Time Jitter RNG	Non-Physical	See Tested Operational Environment Table	256 bits	Full entropy	SHA3-256 (A3034)

Table 12: Entropy Sources

2.9 Key Generation

The bound OpenSSL module provides an SP 800-90A Rev. 1-compliant Deterministic Random Bit Generator (DRBG) for the creation of key components of asymmetric keys, and random number generation.

For generating RSA, ECDSA keys, the bounded module implements asymmetric key generation services compliant with FIPS 186-4. A seed (i.e., the random value) used in asymmetric key generation is directly obtained from the SP 800-90A Rev. 1 DRBG.

The public and private keys used in EC Diffie-Hellman shared secret computation are generated internally by the bound OpenSSL module using the ECDSA key generation method compliant with FIPS 186-4 and SP 800-56A Rev. 3. In accordance with FIPS 140-3 IG D.H, the bound cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per section 5.1 of SP 800-133 Rev. 2 (vendor affirmed) by obtaining a random bit string directly from an approved DRBG and that can support the required security strength requested by the caller (without any V, as described in Additional Comments 2 of IG D.H).

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.10 Key Establishment

The module provides EC Diffie-Hellman shared secret computation compliant with SP 800-56A Rev. 3, in accordance with scenario 2 (1) of IG D.F.

According to Table 2: Comparable strengths in SP 800-57 Part 1 Rev. 5, the key sizes of EC Diffie-Hellman provides the following security strength in the approved mode of operation:

- EC Diffie-Hellman key agreement provides between 128 and 256 bits of encryption strength.

2.11 Industry Protocols

The module does not implement any industry protocols.

© 2025 SUSE, 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

The logical interfaces are the API through which applications request services. The ports and interfaces are shown in the following table.

All data output via data output interface is inhibited when the module is performing pre-operational testing or zeroization, or when the module enters the error state.

No control output is listed as the module does not implement a control output interface.

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API input parameters for data.
N/A	Data Output	API output parameters for data.
N/A	Control Input	API function calls, API input parameters for control input, /proc/sys/crypto/fips_enabled control file, ICAPATH environment variable.
N/A	Status Output	API return codes, API output parameters for status output.

Table 13: Ports and Interfaces

© 2025 SUSE, 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

The module does not support authentication.

4.2 Roles

The module does not support multiple concurrent operators. The Crypto Officer role is implicitly and always assumed by the operator of the module.

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	Crypto Officer	None

Table 14: Roles

4.3 Approved Services

The table below lists the approved services. For each service, the table lists the associated cryptographic algorithm(s), the role to perform the service, the cryptographic keys or CSPs involved, and their access type(s). The following convention is used to specify access rights to a CSP:

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

The details of the approved cryptographic algorithms including the CAVP certificate numbers can be found in Section 2.5. The module implements the ica_get_fips_indicator() API function that outputs the value of .fips_approved field, when "1" is output it, indicates that the service is approved.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric encryption	Perform AES encryption	ica_get_fips_indicator() returns .fips_approved = 1	Plaintext, AES key, IV	Ciphertext	Symmetric encryption	Crypto Officer - AES key: W,E
Symmetric decryption	Perform AES decryption	ica_get_fips_indicator() returns .fips_approved = 1	Ciphertext, AES key, IV	Plaintext	Symmetric decryption	Crypto Officer - AES key: W,E

© 2025 SUSE, 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
Authenticated symmetric encryption	Perform authenticated AES encryption	ica_get_fips_indicator() returns .fips_approved = 1	Plaintext, AES key, IV	Ciphertext, MAC tag	Authenticated symmetric encryption	Crypto Officer - AES key: W,E
Authenticated symmetric decryption	Perform authenticated AES decryption	ica_get_fips_indicator() returns .fips_approved = 1	Ciphertext, MAC tag, AES key, IV	Plaintext or Fail	Authenticated symmetric decryption	Crypto Officer - AES key: W,E
Asymmetric key generation	Generate RSA or ECDSA key pairs	ica_get_fips_indicator() returns .fips_approved = 1	RSA modulus or elliptic curve	RSA or ECDSA key pair	Key pair generation	Crypto Officer - Module-generated ECDSA public key: G,R - Module-generated ECDSA private key: G,R - Module-generated RSA public key: G,R - Module-generated RSA private key: G,R
Digital signature generation	Generate ECDSA signature	ica_get_fips_indicator() returns .fips_approved = 1	Message, ECDSA	Signature	Digital signature generation	Crypto Officer - ECDSA private

© 2025 SUSE, 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
			private key			key: W,E
Digital signature verification	Verify ECDSA signature	ica_get_fips_indicator() returns .fips_approved = 1	Message, Signature, ECDSA public key	Pass or Fail	Digital signature verification	Crypto Officer - ECDSA public key: W,E
Shared secret computation	EC Diffie-Hellman shared secret computation	ica_get_fips_indicator() returns .fips_approved = 1	ECDH private key, ECDH public key from peer	ECDH shared secret	Shared secret computation	Crypto Officer - EC Diffie-Hellman private key: W,E - EC Diffie-Hellman public key from peer: W,E - EC Diffie-Hellman shared secret: G,R
Message digest	Compute SHA hashes	ica_get_fips_indicator() returns .fips_approved = 1	Message	Message digest	Message digest	Crypto Officer
Message authentication code (MAC) generation	Compute AES-based CMAC or GMAC	ica_get_fips_indicator() returns .fips_approved = 1	Message, AES key	MAC tag	Message authentication code (MAC) with AES	Crypto Officer - AES key: W,E
Message authentication code	Verify AES-based	ica_get_fips_indicator() returns .fips_approved = 1	Message, MAC tag, AES key	Pass or Fail	Message authentication code	Crypto Officer - AES

© 2025 SUSE, 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
(MAC) verification	CMAC or GMAC				(MAC) with AES	key: W,E
Show status	Show module status	Implicit (always approved)	None	Module status	None	Crypto Officer
Zeroization	Zeroize CSPs	Implicit (always approved)	Context containing SSPs	None	None	Crypto Officer - AES key: Z - ECDSA public key: Z - ECDSA private key: Z - EC Diffie-Hellman private key: Z - EC Diffie-Hellman public key from peer: Z - EC Diffie-Hellman shared secret: Z - Module-generated ECDSA public key: Z - Module-generated ECDSA

© 2025 SUSE, 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
						private key: Z - Module-generated RSA public key: Z - Module-generated RSA private key: Z - Entropy input: Z - DRBG seed: Z - DRBG internal state (V, key): Z
On-demand Integrity test and self-tests	Perform integrity test and self-tests on-demand	Implicit (always approved)	None	Pass or Fail	Message authentication code (MAC) with HMAC Random number generation Authenticated symmetric encryption Authenticated symmetric decryption Message authentication code (MAC) with AES Symmetric encryption	Crypto Officer

© 2025 SUSE, 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
					Symmetric decryption Digital signature generation Digital signature verification Shared secret computation Message digest	
Self-tests	Perform self-tests	Implicit (always approved)	None	Pass or Fail	Message authentication code (MAC) with HMAC Random number generation Authenticated symmetric encryption Authenticated symmetric decryption Message authentication code (MAC) with AES Symmetric encryption Symmetric decryption Digital signature generation Digital signature verification Shared secret computation	Crypto Officer

© 2025 SUSE, 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 digest	
Show module name and version	Show module name and version	Implicit (always approved)	None	Module name and version	None	Crypto Officer
Module installation and configuration	Install and configure module	Implicit (always approved)	None	Success or Fail	None	Crypto Officer

Table 15: Approved Services

4.4 Non-Approved Services

The table below lists the non-approved services. The details of the non-approved cryptographic algorithms available in the non-approved mode can be found in the Non-Approved Algorithms table in Section 2.5.

Name	Description	Algorithms	Role
RSA modulus exponentiation primitive.	Sign, verify, encrypt, and decrypt primitives.	RSA modulus exponentiation primitive.	CO
DRBG	Random Number Generation service	DRBG	CO

Table 16: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not support the loading of external software/firmware.

© 2025 SUSE, 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 an HMAC-SHA2-256 value calculated at run time with the HMAC value stored in the .hmac file that was computed at build time for each software component of the module. If the HMAC values do not match, the test fails and the module enters the error state. The module uses the HMAC-SHA2-256 algorithm provided by the OpenSSL bound module. The key used for the HMAC calculation is hardcoded in the module.

5.2 Initiate on Demand

Integrity tests are performed as part of the Pre-Operational Self-Tests.

The module provides the Self-Test service to perform self-tests on demand which includes the pre-operational tests (i.e., integrity test) and cryptographic algorithm self-tests (CASTs). This service can be invoked by using the ica_fips_powerup_tests() API function call. During the execution of the on-demand self-tests, services are not available, and no data output or input is possible.

In order to verify whether the self-tests have succeeded and the module is in the Operational state, the calling application may invoke the ica_fips_status(). See Section 10.4 for more information about the possible return values.

© 2025 SUSE, 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

This module operates in a modifiable operational environment per the FIPS 140-3 level 1 specifications. The SUSE Linux Enterprise Server operating system is used as the basis of other products. Compliance is maintained for SUSE products whenever the binary is found unchanged per the vendor affirmation from SUSE based on allowance FIPS 140-3 management manual section 7.9.1 item 1. c) i).

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

Type of Operational Environment: Modifiable

6.2 Configuration Settings and Restrictions

The module shall be installed as stated in Section 11.1. 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 utilities, 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-tested operational environment.

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

7 Physical Security

7.1 Mechanisms and Actions Required

The Libica module inherits the physical characteristics of the host running it; the module has no physical security characteristics of its own. Figure 2 illustrates the IBM System z15 mainframe computer that represents the testing platform, that includes the hardware component of the cryptographic module.



Figure 2 – IBM z15 Mainframe Computer

The Central Processor Assist for Cryptographic Functions (CPACF) is part of the CoProcessor Unit (CoP) integrated in the z15 processor, and offers full implementation of several algorithms; this module uses the processor algorithm implementations (PAI) for AES, SHA-1, SHA-2, SHA-3, SHAKE, ECDSA signature generation and verification, and ECDH shared secret computation. The module is a multi-chip standalone with a physical security level of 1. This security level is satisfied by the device (CoP) being included within the cryptographic boundary of the module and the device being made of production grade components that include standard passivation techniques. The module does not implement a maintenance access interface.

With regards to the CPACF physical design, each microprocessor (core) on the 8-core chip (see Figure 3) has its own dedicated CoP, which implements the crypto instructions and provides the hardware compression function. The compression unit is integrated with the CPACF, benefiting from combining (sharing) the use of buffers and interfaces.

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

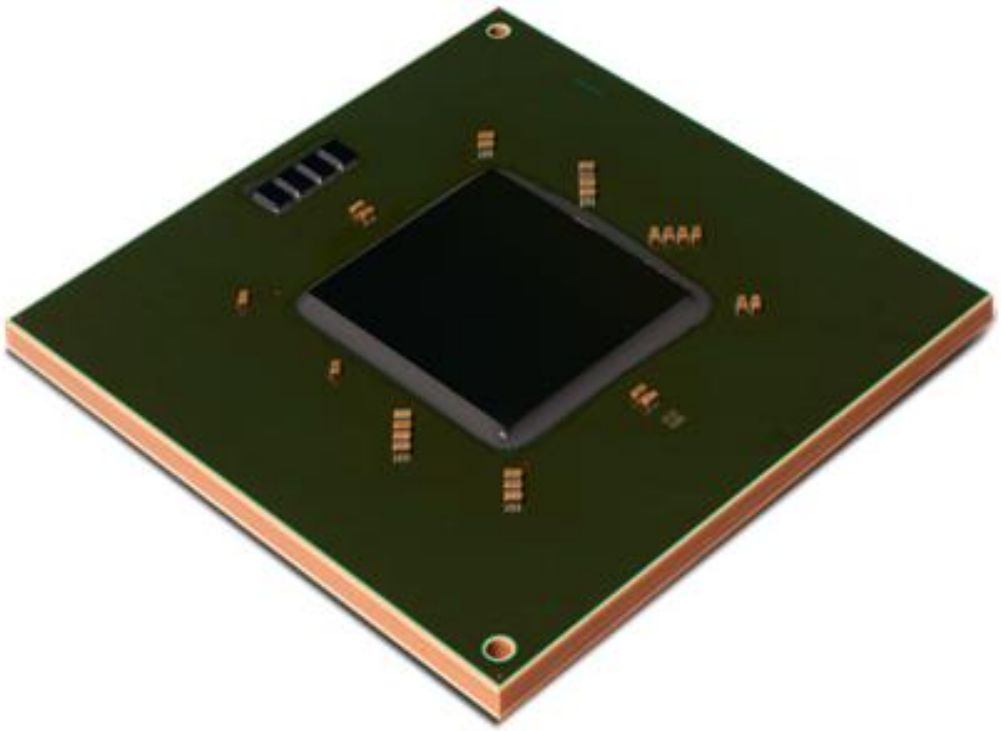


Figure 3 – IBM z15 Processor Unit Chip

© 2025 SUSE, 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.

© 2025 SUSE, 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

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.

Storage Area Name	Description	Persistence Type
RAM	Temporary storage for SSPs used by the module as part of service execution.	Dynamic

Table 17: Storage Areas

9.2 SSP Input-Output Methods

The module does not support direct manual SSP entry or intermediate SSP generation output. The SSPs are provided to the module via API input parameters in plaintext form and output via API output parameters in plaintext form within the physical perimeter of the operational environment. This is allowed by FIPS 140-3 IG 9.5.A, according to the "CM Software to/from App via TOEPP Path" entry on the Key Establishment Table.

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 18: SSP Input-Output Methods

9.3 SSP Zeroization Methods

For those SSPs that are either imported or exported, it is the responsibility of the calling application to zeroize them once they are no longer utilized.

Internal values to store SSPs are zeroized automatically before returning the control to the calling application. Zeroization is performed by overwriting the memory with zeroes.

For services implemented in the bound OpenSSL module, the Libica module calls internally the appropriate zeroization functions provided by the bound module (e.g. OPENSSL_cleanse) before returning to the calling application. The zeroization functions overwrite the memory occupied by SSPs with "zeros" and deallocate the memory with the regular memory deallocation operating system call.

The completion of a zeroization routine will indicate that a zeroization procedure succeeded. Also, module reset can zeroize all SSPs for both the module and the OpenSSL bound module.

© 2025 SUSE, 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
Wipe and Free memory block allocated	Zeroizes the SSPs contained within the cipher handle.	Memory occupied by SSPs is overwritten with zeroes and then it is released, which renders the SSP values irretrievable. The completion of the zeroization routine indicates that the zeroization procedure succeeded.	By calling the cipher related zeroization API
Automatic	Automatically zeroized by the module when no longer needed via calls to the API of the bound OpenSSL module.	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 19: SSP Zeroization Methods

9.4 SSPs

The two tables below summarize the Sensitive Security Parameters (SSPs) that are used by the cryptographic services implemented in the module.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES key	AES key	128, 192, 256 bits - 128, 192, 256 bits	Symmetric key - CSP			Authenticated symmetric encryption Authenticated symmetric decryption Message authentication code (MAC) with AES Symmetric encryption Symmetric decryption
ECDSA public key	ECDSA public key	P-256, P-384, P-521	Public key - PSP			Digital signature verification

© 2025 SUSE, 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, 192, 256 bits				
ECDSA private key	ECDSA private key	P-256, P-384, P-521 - 128, 192, 256 bits	Private key - CSP			Digital signature generation
EC Diffie-Hellman private key	EC Diffie-Hellman private key	P-256, P-384, P-521 - 128, 192, 256 bits	Private key - CSP			Shared secret computation
EC Diffie-Hellman public key from peer	EC Diffie-Hellman public key from peer	P-256, P-384, P-521 - 128, 192, 256 bits	Public key - PSP			Shared secret computation
EC Diffie-Hellman shared secret	EC Diffie-Hellman shared secret	P-256, P-384, P-521 - 128, 192, 256 bits	Shared Secret - CSP		Shared secret computation	
Module-generated ECDSA public key	Module-generated ECDSA public key	P-256, P-384, P-521 - 128, 192, 256 bits	Public key - PSP	Key pair generation		
Module-generated ECDSA private key	Module-generated ECDSA private key	P-256, P-384, P-521 - 128, 192, 256 bits	Private key - CSP	Key pair generation		
Module-generated RSA public key	Module-generated RSA public key	2048, 3072, 4096 bits - 112, 128, 149 bits	Public key - PSP	Key pair generation		
Module-generated RSA private key	Module-generated RSA private key	2048, 3072, 4096 bits - 112, 128, 149 bits	Private key - CSP	Key pair generation		

© 2025 SUSE, 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
Entropy input	Entropy input (IG D.L compliant)	192 to 384 bits - 192 to 384 bits	Entropy Input - CSP	Random number generation		Random number generation
DRBG seed	DRBG seed (IG D.L compliant)	192 to 384 bits - 192 to 256 bits	Seed - CSP	Counter DRBG (A3150)		Random number generation
DRBG internal state (V, key)	DRBG internal state (V, key) (IG D.L compliant)	256, 320, 384 bits - 128, 192, 256 bits	Internal state - CSP	Counter DRBG (A3150)		Random number generation

Table 20: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	
ECDSA public key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	ECDSA private key:Paired With
ECDSA private key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	ECDSA public key:Paired With
EC Diffie-Hellman private key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	EC Diffie-Hellman public key from peer:Used With EC Diffie-Hellman shared secret:Establishes
EC Diffie-Hellman	API input parameters	RAM:Plaintext	From service invocation to	Wipe and Free memory block	EC Diffie-Hellman private key:Used With

© 2025 SUSE, 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
public key from peer			service completion	allocated Module Reset	EC Diffie-Hellman shared secret:Establishes
EC Diffie-Hellman shared secret	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	EC Diffie-Hellman private key:Established by EC Diffie-Hellman public key from peer:Established by
Module-generated ECDSA public key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Module-generated ECDSA private key:Paired With
Module-generated ECDSA private key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Module-generated ECDSA public key:Paired With
Module-generated RSA public key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Module-generated RSA private key:Paired With
Module-generated RSA private key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated	Module-generated RSA public key:Paired With
Entropy input		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Automatic Module Reset	DRBG seed:Derives
DRBG seed		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Automatic Module Reset	Entropy input:Derived From DRBG internal state (V, key):Derives

© 2025 SUSE, 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
DRBG internal state (V, key)		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Automatic Module Reset	DRBG seed:Derived From

Table 21: SSP Table 2

9.5 Transitions

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

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

10 Self-Tests

The module performs pre-operational tests and CASTs automatically when the module is loaded into memory. Pre-operational tests ensure that the module is not corrupted, and the CASTs ensure that the cryptographic algorithms work as expected. While the module is executing the pre-operational tests and CASTs, services are not available, and input and output are inhibited. The module is not available for use by the calling application until the pre-operational tests are completed successfully. After the pre-operational test and the CASTs succeed, the module becomes operational. If any of the pre-operational test or any of the CASTs fail an error message is returned, and the module transitions to the error state.

10.1 Pre-Operational Self-Tests

The module performs the integrity test of the software component using HMAC-SHA2-256. The integrity test uses the HMAC algorithm implemented in the bound OpenSSL module, which tests this algorithm as part of the CASTs when loaded into memory and before Libica performs the pre-operational tests. The details of the integrity test are provided in Section 5.1.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A3147)	SHA2-256	MAC Tag Verification	SW/FW Integrity	Module becomes operational	N/A

Table 22: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

The following table specifies the CASTs performed by the Libica module as well as the Pair-wise Consistency Tests (PCTs) performed by the bound OpenSSL module.

All CASTs performed are in the form of the Known Answer Tests (KATs) and are run prior to performing the integrity test. A KAT includes the comparison of a calculated output with an expected known answer, hard coded as part of the test vectors used in the test. If the values do not match, the KAT fails.

If any PCT or CAST fails, the module returns an error code and enters the Error state. When the module is in the Error state, no data is output and cryptographic operations are not allowed.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A3378) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3378) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on

© 2025 SUSE, 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
						before the integrity test
AES-CBC (A3378) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A3378) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC-CS1 (A3378) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC-CS1 (A3378) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC-CS2 (A3378) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC-CS2 (A3378) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC-CS3 (A3378) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC-CS3 (A3378) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test

© 2025 SUSE, 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
AES-OFB (A3378) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-OFB (A3378) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3378) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3378) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CCM (A3378) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CCM (A3378) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A3377) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A3378) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A3377) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test

© 2025 SUSE, 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
AES-GCM (A3378) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CMAC (A3378) - Encryption	128-bit key, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-GMAC (A3377) - Encryption	128-bit key, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-GMAC (A3378) - Encryption	128-bit key, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A3378)	P-256 curve	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3378)	P-256 with SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3378)	P-256 with SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
SHA-1 (A3378)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3378)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test

© 2025 SUSE, 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
SHA2-256 (A3378)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3378)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3378)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-224 (A3378)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A3378)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-384 (A3378)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-512 (A3378)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3147)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication code (performed by bound OpenSSL module)	Test runs at power-on before the integrity test
SHA2-256 (A3147)		KAT	CAST	Module becomes operational	Message digest (performed by bound OpenSSL module)	Test runs at power-on before the integrity test

© 2025 SUSE, 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
ECDSA KeyGen (FIPS186-4) (A3147)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification (performed by bound OpenSSL module)	Key pair generation
RSA KeyGen (FIPS186-4) (A3147)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification (performed by bound OpenSSL module)	Key pair generation
Counter DRBG (A3150)	AES with 256-bit key, with/without DF, with/without PR	KAT	CAST	Module becomes operational	Random number generation (performed by bound OpenSSL module)	Test runs at power-on before the integrity test

Table 23: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A3147)	MAC Tag Verification	SW/FW Integrity	On Demand	Manually

Table 24: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A3378) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A3378) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC-CS1 (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC-CS1 (A3378) - Decryption	KAT	CAST	On Demand	Manually

© 2025 SUSE, 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
AES-CBC-CS2 (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC-CS2 (A3378) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A3378) - Decryption	KAT	CAST	On Demand	Manually
AES-OFB (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-OFB (A3378) - Decryption	KAT	CAST	On Demand	Manually
AES-CTR (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-CTR (A3378) - Decryption	KAT	CAST	On Demand	Manually
AES-CCM (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-CCM (A3378) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A3377) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A3377) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A3378) - Decryption	KAT	CAST	On Demand	Manually
AES-CMAC (A3378) - Encryption	KAT	CAST	On Demand	Manually
AES-GMAC (A3377) - Encryption	KAT	CAST	On Demand	Manually
AES-GMAC (A3378) - Encryption	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A3378)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3378)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3378)	KAT	CAST	On Demand	Manually
SHA-1 (A3378)	KAT	CAST	On Demand	Manually
SHA2-224 (A3378)	KAT	CAST	On Demand	Manually
SHA2-256 (A3378)	KAT	CAST	On Demand	Manually

© 2025 SUSE, 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
SHA2-384 (A3378)	KAT	CAST	On Demand	Manually
SHA2-512 (A3378)	KAT	CAST	On Demand	Manually
SHA3-224 (A3378)	KAT	CAST	On Demand	Manually
SHA3-256 (A3378)	KAT	CAST	On Demand	Manually
SHA3-384 (A3378)	KAT	CAST	On Demand	Manually
SHA3-512 (A3378)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3147)	KAT	CAST	On Demand	Manually
SHA2-256 (A3147)	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3147)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3147)	PCT	PCT	On Demand	Manually
Counter DRBG (A3150)	KAT	CAST	On Demand	Manually

Table 25: Conditional Periodic Information

10.4 Error States

When the module fails any pre-operational self-test or conditional test, the module will return an error code to indicate the error and will enter the Error state. Any further cryptographic operation is inhibited. The calling application can obtain the module state by calling the ica_fips_status() API function.

Name	Description	Conditions	Recovery Method	Indicator
Self-test Error State	Error state entered due to self-test failure	Failure of any CAST Failure of any integrity test Bound OpenSSL module transitions to error state	Restart module	All cryptographic operations are inhibited, and condition-specific error codes are returned. CAST failure: ICA_FIPS_CRYPTOALG; integrity test failure: ICA_FIPS_INTEGRITY

Table 26: Error States

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

10.5 Operator Initiation of Self-Tests

On-Demand self-tests can be invoked by calling the fips_powerup_tests() API function, which causes the module to run the pre-operational tests again. During the execution of the on-demand self-tests, services are not available, and no data output or input is possible.

To verify whether the self-tests have succeeded, and the module is in the Operational state, the calling application may invoke the ica_fips_status() API function.

© 2025 SUSE, 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

11.1.1 Module Installation

The Crypto Officer can install the RPM packages containing the module as listed in Section 11.1.3 using the zypper tool as follows.

```
# zypper install libica4
```

The integrity of the RPM package is automatically verified during the installation, and the Crypto Officer shall not install the RPM package if there is any integrity error.

11.1.2 Operating Environment Configuration

The module requires to run on a z15 processor that has Feature Code 3863 enabled. This feature enables processor acceleration implementations (PAI) in CPACF for AES and ECC algorithms (SHA-1, SHA-2, SHA-3 and SHAKE algorithms are enabled by default in z15 processors). Make sure with your system administrator that the hardware platform meets this requirement.

In addition, the operating environment needs to be configured to support the approved mode of operation, so the following steps shall be performed with the root privilege:

1. Install the dracut-fips RPM package:

```
# zypper install dracut-fips
```

2. Recreate the INITRAMFS image:

```
# dracut -f
```

3. After regenerating the initrd, the Crypto Officer has to append the following parameter in the /etc/default/grub configuration file in the GRUB_CMDLINE_LINUX_DEFAULT line:

```
fips=1
```

4. After editing the configuration file, please run the following command to change the setting in the boot loader:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

If /boot or /boot/efi resides on a separate partition, the kernel parameter boot=<partition of /boot or /boot/efi> must be supplied. The partition can be identified with the command "df /boot" or "df /boot/efi" respectively. For example:

```
# df /boot
Filesystem     1K-blocks     Used     Available     Use%     Mounted on
/dev/sda1      233191        30454    190296        14%              /boot
```

The partition of /boot is located on /dev/sda1 in this example. Therefore, the following string needs to be appended in the aforementioned grub file:

```
"boot=/dev/sda1"
```

5. Reboot to apply these settings.

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Now, the operating environment is configured to support the approved mode of operation. The Crypto Officer should check the existence of the file /proc/sys/crypto/fips_enabled, and verify it contains a numeric value "1". If the file does not exist or does not contain "1", the operating environment is not configured to support the approved mode of operation and the module will not operate as a FIPS validated module properly.

11.1.3Module Installation for Vendor Affirmed Platforms

The following table includes the information on module installation process for the vendor affirmed platforms that are listed in Section 2.2.

Product	Link
SUSE Linux Enterprise Micro 5.3	https://documentation.suse.com/sle-micro/5.3/single-html/SLE-Micro-security/#sec-fips-slemicro-install
SUSE Linux Enterprise Base Container Image 15SP4	https://documentation.suse.com/smart/linux/html/concept-bci/index.html

Table 27: Installation for Vendor Affirmed Platforms

Note: Per section 7.9 of the FIPS 140-3 Management Manual, the Cryptographic Module Validation Program (CMVP) makes no statement as to the correct operation of the module or the security strengths of the generated keys when this module is ported and executed in an operational environment not listed on the validation certificate.

11.2 Administrator Guidance

The binaries of the module are contained in the RPM packages for delivery. The Crypto Officer shall follow sections 11.1.1 and 11.1.2 to configure the operational environment and install the module to be operated as a FIPS 140-3 validated module.

The table below lists the RPM package that contains the FIPS validated module and the OE directory where the components are installed. The "Show module name and version" service is implemented via the following API functions:

- ica_get_build_version() returns the software version, whose value is "FIPS-SUSE-4.2.1-150400.3.8.1". This identifier matches the version included in the RPM package filenames.
- ica_get_hw_info() returns the hardware version, whose value is "IBM/S390, model type 8561". This identifier corresponds to the IBM z/15 hardware with z15 processor.

Processor Architecture	RPM Packages
z15	libica-4.2.1-150400.3.8.1.s390x.rpm

Table 28: RPM packages

11.3 Non-Administrator Guidance

The module has no non-administrator guidance.

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

11.4 End of Life

For secure sanitization of the cryptographic module, the module must first to be powered off, which will zeroize all keys and CSPs in volatile memory. Then, for actual deprecation, the module shall be upgraded to a newer version that is FIPS 140-3 validated.

The module does perform persistent storage of SSPs, so further sanitization steps are not required.

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

12 Mitigation of Other Attacks

The module does not implement any mitigation mechanism.

© 2025 SUSE, 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
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CPACF	Central Processor Assist for Cryptographic Function
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DF	Derivation Function
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
KAS	Key Agreement Schema
KAT	Known Answer Test
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
O/S	Operating System
PAA	Processor Algorithm Acceleration
PAI	Processor Algorithm Implementation
PR	Prediction Resistance
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

SHS	Secure Hash Standard
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Appendix B. References

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** October 2022 https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
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 800-38B	**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
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 of Storage Devices** January 2010 https://doi.org/10.6028/NIST.SP.800-38E
FIPS 180-4	**Secure Hash Standard (SHS)** August 2015 https://doi.org/10.6028/NIST.FIPS.180-4
FIPS 202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 https://doi.org/10.6028/NIST.FIPS.202
FIPS 198-1	**The Keyed-Hash Message Authentication Code (HMAC)** July 2008 https://doi.org/10.6028/NIST.FIPS.198-1

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

FIPS 186-4	**Digital Signature Standard (DSS)** July 2013 https://doi.org/10.6028/NIST.FIPS.186-4
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 800-90A 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
SP 800-133 Rev. 2	**Recommendation for Cryptographic Key Generation** June 2020 https://doi.org/10.6028/NIST.SP.800-133r2
SP 800-57 Part 1 Rev. 5	**Recommendation for Key Management: Part 1 - General** May 2020 https://doi.org/10.6028/NIST.SP.800-57pt1r5

© 2025 SUSE, LLC / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.