SUSE LLC

SUSE Linux Enterprise Libica Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Prepared by:

atsec information security corporation
4516 Seton Center Pkwy, Suite 250
Austin, TX 78759
www.atsec.com

Document version: 1.2
Last update: 2026-09-06

© 2026 SUSE LLC/atsec information security corporation.

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

Table of Contents

1 General 7
1.1 Overview 7
1.2 Security Levels 7
1.3 Additional Information 8
2 Cryptographic Module Specification 9
2.1 Description 9
2.2 Tested and Vendor Affirmed Module Version and Identification 10
2.3 Excluded Components 11
2.4 Modes of Operation 12
2.5 Algorithms 12
2.6 Security Function Implementations 16
2.7 Algorithm Specific Information 18
2.7.1 AES XTS 18
2.7.2 AES GCM IV 19
2.7.3 SP 800-56A Rev. 3 Assurances 19
2.8 RBG and Entropy 19
2.9 Key Generation 20
2.10 Key Establishment 20
2.11 Industry Protocols 20
3 Cryptographic Module Interfaces 21
3.1 Ports and Interfaces 21
4 Roles, Services, and Authentication 22
4.1 Authentication Methods 22
4.2 Roles 22
4.3 Approved Services 22
4.4 Non-Approved Services 27
4.5 External Software/Firmware Loaded 27
5 Software/Firmware Security 29
5.1 Integrity Techniques 29

© 2026 SUSE LLC/atsec information security corporation.

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

5.2 Initiate on Demand 29
6 Operational Environment 30
6.1 Operational Environment Type and Requirements 30
6.2 Configuration Settings and Restrictions 30
7 Physical Security 31
8 Non-Invasive Security 33
9 Sensitive Security Parameters Management 34
9.1 Storage Areas 34
9.2 SSP Input-Output Methods 34
9.3 SSP Zeroization Methods 34
9.4 SSPs 35
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 45
10.4 Error States 48
10.5 Operator Initiation of Self-Tests 49
11 Life-Cycle Assurance 50
11.1 Installation, Initialization, and Startup Procedures 50
11.2 Administrator Guidance 50
11.3 Non-Administrator Guidance 51
11.4 Design and Rules 51
11.5 Maintenance Requirements 51
11.6 End of Life 51
12 Mitigation of Other Attacks 52
Appendix A. Glossary and Abbreviations 53
Appendix B. References 55

© 2026 SUSE LLC/atsec information security corporation.

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

© 2026 SUSE LLC/atsec information security corporation.

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) 10
Table 3: Tested Operational Environments - Software, Firmware, Hybrid 10
Table 4: Tested Module Identification – Hybrid Disjoint Hardware 11
Table 5: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid 11
Table 6: Modes List and Description 12
Table 7: Approved Algorithms - 14
Table 8: Approved Algorithms - [EVM] 15
Table 9: Vendor-Affirmed Algorithms 15
Table 10: Non-Approved, Not Allowed Algorithms 16
Table 11: Security Function Implementations 18
Table 12: Entropy Certificates 19
Table 13: Entropy Sources 19
Table 14: Ports and Interfaces 21
Table 15: Roles 22
Table 16: Approved Services 27
Table 17: Non-Approved Services 27
Table 18: Storage Areas 34
Table 19: SSP Input-Output Methods 34
Table 20: SSP Zeroization Methods 35
Table 21: SSP Table 1 37
Table 22: SSP Table 2 38
Table 23: Pre-Operational Self-Tests 39
Table 24: Conditional Self-Tests 45
Table 25: Pre-Operational Periodic Information 45
Table 26: Conditional Periodic Information 48
Table 27: Error States 48

List of Figures

© 2026 SUSE LLC/atsec information security corporation.

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

Figure 1: Block Diagram 10
Figure 2: IBM Mainframe Computer 31
Figure 3: IBM® Telum™ Processor Unit Chip 32

© 2026 SUSE LLC/atsec information security corporation.

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 1.2 and 1.3 and hardware version IBM z/16 of the SUSE Linux Enterprise Libica Cryptographic Module. 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.

1.2 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

© 2026 SUSE LLC/atsec information security corporation.

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

1.3 Additional Information

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.

© 2026 SUSE LLC/atsec information security corporation.

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 and a Central Processor Assist for Cryptographic Functions (CPACF). The software library provides the API and a subset of the cryptographic algorithms. The CPACF is a hardware device that is part of the IBM® Telum™ processor. It provides additional cryptographic algorithms implemented in hardware. In addition, the module uses a bound OpenSSL module, which provides additional algorithms. The Libica module is tested with versions 1.2 and 1.3 which bound to "SUSE Linux Enterprise OpenSSL 1 Cryptographic Module" with certificate 5238 and "SUSE Linux Enterprise OpenSSL 3 Cryptographic Module" with certificate 5096, respectively.

Module Type: Software-hybrid

Module Embodiment: MultiChipStand

Cryptographic Boundary:

The blue dashed line in the block diagram below shows the cryptographic boundary of the module and its interfaces with the operational environment. The cryptographic boundary of the module is composed of the Libica shared library, the .hmac check file for this shared library, and the CPACF hardware implementation provided by the IBM® Telum™ processor that is installed in the TOEPP where the module is executing.

The TOEPP of the module of the module is the IBM Mainframe on which the IBM® Telum™ processor is installed.

© 2026 SUSE LLC/atsec information security corporation.

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

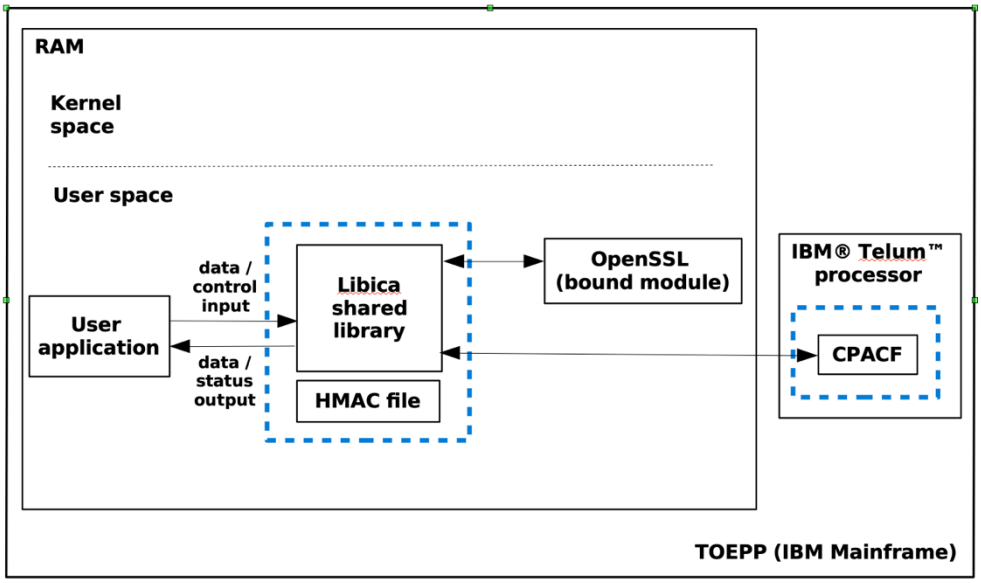

Figure 1: Block Diagram

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
libica-openssl1_1.so.4.3.1, .libica-openssl1_1.so.4.3.1.hmac	1.2	N/A	HMAC-SHA-256
libica.so.4.3.1, .libica.so.4.3.1.hmac	1.3	N/A	HMAC-SHA-256

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

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
SUSE Linux Enterprise Server 15 SP6	IBM z16 A01	IBM® Telum(TM)	Yes	N/A	1.2 1.3

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

© 2026 SUSE LLC/atsec information security corporation.

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

Tested Module Identification – Hybrid Disjoint Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
IBM® Telum(TM)	IBM z16	N/A	IBM® Telum(TM)	CPACF

Table 4: Tested Module Identification – Hybrid Disjoint Hardware

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

Operating System	Hardware Platform
SUSE Linux Enterprise Server 15SP6	IBM LinuxONE III Model LT1 on z15
SUSE Linux Enterprise Base Container Image 15SP6	IBM LinuxONE III Model LT1 on z15
SUSE Linux Enterprise Base Container Image 15SP6	IBM LinuxONE III Model LT1 QEMU VM on z15
SUSE Linux Enterprise Base Container Image 15SP6	IBM z16 A01 on IBM® Telum(TM)
SUSE Linux Enterprise Server 15SP6	IBM LinuxONE III Model LT1 QEMU VM on z15
SUSE Linux Enterprise Server 15SP7	IBM LinuxONE III Model LT1 on z15
SUSE Linux Enterprise Server 15SP7	IBM z16 A01 on IBM® Telum(TM)
SUSE Linux Enterprise Base Container Image 15SP7	IBM LinuxONE III Model LT1 on z15
SUSE Linux Enterprise Base Container Image 15SP7	IBM z16 A01 on IBM® Telum(TM)

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

The module is considered to maintain compliance with the FIPS 140-3 validation for SUSE products when operating on any general-purpose platform/processor that supports the SUSE Linux Enterprise Server operating system per the vendor affirmation from SUSE based on the allowance FIPS 140-3 management manual [FIPS140-3_MM] section 7.9.1 bullet 1 a i).

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 excluded from the requirements of the FIPS 140-3 standard.

© 2026 SUSE LLC/atsec information security corporation.

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

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	Automatically entered whenever an approved service is requested	Approved	Equivalent to the indicator of the requested service (the API for the requested service returns value 0)
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Equivalent to the indicator of the requested service (the API for the requested service returns non zero value)

Table 6: 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	A6712	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A6712	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A6712	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

© 2026 SUSE LLC/atsec information security corporation.

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

Algorithm	CAVP Cert	Properties	Reference
AES-CMAC	A6712	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A6712	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A6712	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A6659, A6711, A6712	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.2 Key Length - 128, 192, 256	SP 800-38D
AES-GMAC	A6659, A6711, A6712	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.2 Key Length - 128, 192, 256	SP 800-38D
AES-OFB	A6712	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A6712	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
ECDSA SigGen (FIPS186-5)	A6712	Component - Yes	FIPS 186-5
KAS-ECC-SSC Sp800-56Ar3	A6712	Domain Parameter Generation Methods - P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
SHA-1	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-224	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-256	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4

© 2026 SUSE LLC/atsec information security corporation.

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

Algorithm	CAVP Cert	Properties	Reference
SHA2-384	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512/224	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512/256	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA3-224	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 202
SHA3-256	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 202
SHA3-384	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 202
SHA3-512	A6712	Message Length - Message Length: 8-65536 Increment 8	FIPS 202
SHAKE-128	A6712	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A6712	Output Length - Output Length: 16-65536 Increment 8	FIPS 202

Table 7: Approved Algorithms -

[EVM]

Algorithm	CAVP Cert	Properties	Reference
Counter DRBG	A5397, A5659	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - No, Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A5883, A6103	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5

© 2026 SUSE LLC/atsec information security corporation.

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

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-256	A5883, A6103	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
RSA KeyGen (FIPS186-5)	A5883	Key Generation Mode - probableWithProbableAux Modulo - 2048, 3072, 4096, 6144, 8192 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA KeyGen (FIPS186-5)	A6103	Key Generation Mode - probable Modulo - 2048, 3072, 4096, 6144, 8192 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
SHA2-256	A5883, A6103	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Table 8: Approved Algorithms - [EVM]

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
Asymmetric Cryptographic Key Generation (CKG)	Key type:Asymmetric Provided by:[EVM]	N/A	SP 800-133 Rev. 2, section 4, example 1

Table 9: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES GCM with external IV (Libica)	Authenticated symmetric encryption

© 2026 SUSE LLC/atsec information security corporation.

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

Name	Use and Function
RSA (Libica)	Signature generation, signature verification, encryption, and decryption primitives
DRBG (Libica)	Random Number Generation service
ECDSA (Libica)	Signature verification component

Table 10: 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 Non-Approved Services.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric encryption	BC-UnAuth	Encrypt a message with AES		AES-CBC: (A6712) AES-CFB128: (A6712) AES-CFB8: (A6712) AES-CTR: (A6712) AES-ECB: (A6712) AES-OFB: (A6712) AES-XTS Testing Revision 2.0: (A6712)
Symmetric decryption	BC-UnAuth	Decrypt a message with AES		AES-CBC: (A6712) AES-CFB128: (A6712) AES-CFB8: (A6712) AES-CTR: (A6712) AES-ECB: (A6712) AES-OFB: (A6712) AES-XTS Testing Revision 2.0: (A6712)

© 2026 SUSE LLC/atsec information security corporation.

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

Name	Type	Description	Properties	Algorithms
Authenticated encryption	BC-Auth	Encrypt and authenticate a message with AES		AES-GCM: (A6659, A6711, A6712)
Authenticated decryption	BC-Auth	Decrypt and authenticate a message with AES		AES-GCM: (A6659, A6711, A6712)
Message authentication with AES	MAC	Authenticate a message with AES		AES-GMAC: (A6659, A6711, A6712) AES-CMAC: (A6712)
Digital signature generation component with ECDSA	DigSig-SigGen	Generate a signature using ECDSA		ECDSA SigGen (FIPS186-5): (A6712)
Shared secret computation with ECDH	KAS-SSC	Compute a shared secret using ECDH	Compliance:FIPS 140-3 IG D.F, Scenario 2(1)	KAS-ECC-SSC Sp800-56Ar3: (A6712)
Message digest	SHA XOF	Compute the digest of a message using SHA or XOF functions		SHA-1: (A6712) SHA2-224: (A6712) SHA2-256: (A6712) SHA2-384: (A6712) SHA2-512: (A6712) SHA2-512/224: (A6712) SHA2-512/256: (A6712) SHA3-224: (A6712) SHA3-256: (A6712) SHA3-384: (A6712) SHA3-512: (A6712) SHAKE-128: (A6712) SHAKE-256: (A6712)

© 2026 SUSE LLC/atsec information security corporation.

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

Name	Type	Description	Properties	Algorithms
Random number generation	DRBG	[EVM]Generate random numbers	Use:Internal Provided by:Bound OpenSSL module	Counter DRBG: (A5397, A5659)
Key pair generation with ECDSA	AsymKeyPair-KeyGen	[EVM]Generate a key pair using ECDSA	Mode:FIPS 186-5, Section A.2.2 - Rejection Sampling Provided by:Bound OpenSSL module	ECDSA KeyGen (FIPS186-5): (A5883, A6103)
Key pair generation with RSA	AsymKeyPair-KeyGen	[EVM]Generate a key pair using RSA	Mode:FIPS 186-5, Section A.1.6 - Probable Primes Based on Auxiliary Probable Provided by:Bound OpenSSL module	RSA KeyGen (FIPS186-5): (A5883, A6103)
Message authentication with HMAC	MAC	[EVM]Authenticate a message using HMAC	Provided by:Bound OpenSSL module	HMAC-SHA2-256: (A5883, A6103) SHA2-256: (A5883, A6103)

Table 11: 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 generate symmetric keys, the check is performed when keys are input the service APIs.

Key_1 and Key_2 shall 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.

© 2026 SUSE LLC/atsec information security corporation.

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

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 at its entirety randomly. The DRBG provided by the OpenSSL bound module, compliant with SP 800-90A Rev. 1, is used for generating the IV of 96 bits. 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 within its physical perimeter.

2.7.3 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 in the context of the TLS or SSH protocols. Additionally, the module's approved key pair generation service (see Approved Services table in Section 4.3 Approved Services) must be used to generate ephemeral EC Diffie-Hellman key pairs. As part of this service, the module will internally perform the full public key validation of the generated public key. Alternatively key pairs may be obtained from another FIPS-validated module.

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

2.8 RBG and Entropy

Cert Number	Vendor Name
E177	SUSE LLC
E209	SUSE LLC

Table 12: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
SUSE OpenSSL CPU Time Jitter RNG Entropy Source	Non-Physical	SUSE Linux Enterprise Server 15 SP6 on IBM® TelumTM	256 bits	256 bits	AES-256-CTR-DRBG (A5397); SHA3-256 (A5411)
SUSE OpenSSL1 CPU Time Jitter RNG Entropy Source	Non-Physical	SUSE Linux Enterprise Server 15 SP6 on IBM® TelumTM	256 bits	256 bits	AES-256-CTR-DRBG (A5555); SHA3-256 (A5411)

Table 13: Entropy Sources

The Deterministic Random Bit Generator (DRBG) is provided by the bound OpenSSL module [EVM]. The DRBG is based on SP 800-90A Rev. 1 for the creation of RSA and ECDSA keys in OpenSSL, and for the random

© 2026 SUSE LLC/atsec information security corporation.

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

generation of the IV used in AES GCM and the k value used in ECDSA signature generation component. The use of the DRBG provided by the bound OpenSSL module is only internal to the Libica module. 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 Table above. This entropy source is located within the physical perimeter, but outside of the cryptographic boundary of the module as well as the EVM. 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.

2.9 Key Generation

For generating RSA, ECDSA keys, the bound OpenSSL module implements asymmetric key generation services compliant with FIPS 186-5. A seed (i.e., the random value) used in asymmetric key generation is directly obtained from the SP 800-90A Rev. 1 OpenSSL's 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 186-5 and SP 800-56A Rev. 3.

2.10 Key Establishment

The module implements shared secret computation method as listed in the Security Function Implementations table in Section 2.6 Security Function Implementations.

2.11 Industry Protocols

The module does not implement any industry protocols.

© 2026 SUSE LLC/atsec information security corporation.

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 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.
N/A	Power	N/A

Table 14: 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 SUSE LLC/atsec information security corporation.

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 methods.

4.2 Roles

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

Table 15: Roles

The module does not support multiple concurrent operators.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric encryption	Perform AES encryption	ica_aes_* return 0	Plaintext, AES key, IV	Ciphertext	Symmetric encryption	Crypto Officer - AES key: W,E
Symmetric decryption	Perform AES decryption	ica_aes_* return 0	Ciphertext, AES key, IV	Plaintext	Symmetric decryption	Crypto Officer - AES key: W,E
Authenticated encryption	Perform authenticated AES encryption	ica_aes_gcm_kma_* return 0	Plaintext, AES key, IV	Ciphertext, MAC tag	Authenticated encryption	Crypto Officer - AES key: W,E
Authenticated decryption	Perform authenticated AES decryption	ica_aes_gcm_kma_* return 0	Ciphertext, MAC tag, AES key, IV	Plaintext or Fail	Authenticated decryption	Crypto Officer - AES key: W,E
Digital signature	Generate ECDSA signature	ica_ecdsa_sign returns 0	Hashed message, ECDSA	Signature	Digital signature generation	Crypto Officer - ECDSA

© 2026 SUSE LLC/atsec information security corporation.

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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
generation component			private key		component with ECDSA Random number generation	private key: W,E
Shared secret computation	EC Diffie-Hellman shared secret computation	ica_ecdh_derive_secret returns 0	ECDH private key, ECDH public key from peer	ECDH shared secret	Shared secret computation with ECDH	Crypto Officer - EC Diffie-Hellman private key: W,E - EC Diffie-Hellman public key: W,E - EC Diffie-Hellman shared secret: G,R
Message digest	Compute SHA or XOF hashes	ica_sha* return 0	Message	Message digest	Message digest	Crypto Officer
Message authentication code (MAC) with AES	Compute AES-based CMAC or GMAC	ica_aes_gcm_kma_* return 0	Message, AES key	MAC tag	Message authentication with AES	Crypto Officer - AES key: W,E
[EVM]Asymmetric key generation	Generate RSA or ECDSA key pairs relying on the bound	ica_ec_key_generate or ica_rsa_key_generate_mod_expo returns 0	RSA modulus length or EC curve	RSA or ECDSA key pair	Key pair generation with ECDSA Key pair	Crypto Officer - [EVM]Bound-module-

© 2026 SUSE LLC/atsec information security corporation.

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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	OpenSSL module				generation with RSA	generated EC public key: G,R - [EVM]Bound-module-generated EC private key: G,R - [EVM]Bound-module-generated RSA public key: G,R - [EVM]Bound-module-generated RSA private key: G,R
Module installation and configuration	Install and configure module	Implicit (always approved)	None	Success or Fail	None	Crypto Officer
Show status	Show module status	Implicit (always approved)	None	Module status	None	Crypto Officer
Show module name and version	Show module name and version	Implicit (always approved)	None	Module name and version	None	Crypto Officer

© 2026 SUSE LLC/atsec information security corporation.

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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Self-tests	Perform self-tests	Implicit (always approved)	None	Pass or Fail	Random number generation Authenticated encryption Authenticated decryption Message authentication with AES Message authentication with HMAC Symmetric encryption Symmetric decryption Digital signature generation component with ECDSA Shared secret computation with ECDH Message digest	Crypto Officer
Zeroization	Zeroize CSPs	Implicit (always approved)	Context containing SSPs	None	None	Crypto Officer - AES key: Z - ECDSA private

© 2026 SUSE LLC/atsec information security corporation.

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 - EC Diffie-Hellman private key: Z - EC Diffie-Hellman public key: Z - EC Diffie-Hellman shared secret: Z - [EVM]Bound-module-generated EC public key: Z - [EVM]Bound-module-generated EC private key: Z - [EVM]Bound-module-generated RSA public key: Z - [EVM]Bound-module-

© 2026 SUSE LLC/atsec information security corporation.

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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						generated RSA private key: Z

Table 16: 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.
- **N/A**: The module does not access any SSP or key during its operation.

The indicator that a service is approved is the return value of the API(s) that are used to invoke the service. A return value of "0" indicates that the service is approved. Any other return value (e.g., EPERM, EACCESS, or EINVAL) indicates that the invoked service is non-approved.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
AES GCM with external IV	Perform authenticated encryption with an externally provided IV	AES GCM with external IV (Libica)	CO
RSA	Sign/verify and encrypt/decrypt primitives	RSA (Libica)	CO
DRBG	Random Number Generation service	DRBG (Libica)	CO
ECDSA	Signature verification component	ECDSA (Libica)	CO

Table 17: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not load external software or firmware.

© 2026 SUSE LLC/atsec information security corporation.

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

© 2026 SUSE LLC/atsec information security corporation.

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 MAC key 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().

© 2026 SUSE LLC/atsec information security corporation.

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.

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.

6.2 Configuration Settings and Restrictions

The module shall be installed as stated in Section 11 Life-Cycle Assurance.

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.

© 2026 SUSE LLC/atsec information security corporation.

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

7 Physical Security

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 z16 mainframe computer that represents the testing platform, that includes the hardware component of the cryptographic module.

Figure 2: IBM Mainframe Computer

The Central Processor Assist for Cryptographic Functions (CPACF) is part of the CoProcessor Unit (CoP) integrated in the IBM® Telum™ 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.

© 2026 SUSE LLC/atsec information security corporation.

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

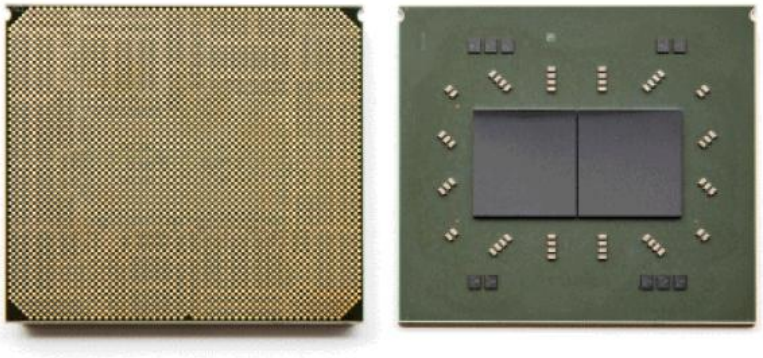

Figure 3: IBM® Telum™ Processor Unit Chip

© 2026 SUSE LLC/atsec information security corporation.

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 mechanisms, and therefore this section is not applicable.

© 2026 SUSE LLC/atsec information security corporation.

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. SSPs are stored until they are zeroized by the operator (using a zeroization call or removing power from the module) or zeroized automatically	Dynamic

Table 18: Storage Areas

The module does not perform persistent storage of SSPs. The SSPs are temporarily stored in the RAM in plaintext form.

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 19: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Zeroized by calling application	Calling application zeroizes the SSPs contained within the cipher handle	By calling the appropriate zeroization functions	By calling the cipher related zeroization API
Automatic	The module zeorizes internal values used to store SSPs and calls OpenSSL	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable. The successful	N/A

© 2026 SUSE LLC/atsec information security corporation.

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

Zeroization Method	Description	Rationale	Operator Initiation
	APIs to zeroize OpenSSL SSP structures	completion of the running service indicates that zeroization has completed	
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 20: 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.

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 input to the module	128, 192, 256 bits - 128, 192, 256 bits	Symmetric key - CSP			Symmetric encryption Symmetric decryption Authenticated encryption Authenticated decryption Message authentication with AES

© 2026 SUSE LLC/atsec information security corporation.

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
ECDSA private key	ECDSA private key input to the module	P-256, P-384, P-521 - 128, 192, 256 bits	Private key - CSP			Digital signature generation component with ECDSA
EC Diffie-Hellman private key	EC Diffie-Hellman private key input to the module	P-256, P-384, P-521 - 128, 192, 256 bits	Private key - CSP			Shared secret computation with ECDH
EC Diffie-Hellman public key	EC Diffie-Hellman public key input to the module	P-256, P-384, P-521 - 128, 192, 256 bits	Public key - PSP			Shared secret computation with ECDH
EC Diffie-Hellman shared secret	EC Diffie-Hellman shared secret established by the module	P-256, P-384, P-521 - 128, 192, 256 bits	Shared Secret - CSP		Shared secret computation with ECDH	
[EVM]Bound-module-generated EC public key	EC public key generated by the bound OpenSSL module	P-256, P-384, P-521 - 128, 192, 256 bits	Public key - PSP	Key pair generation with ECDSA		
[EVM]Bound-module-generated EC private key	EC private key generated by the bound OpenSSL module	P-256, P-384, P-521 - 128, 192, 256 bits	Private key - CSP	Key pair generation with ECDSA		

© 2026 SUSE LLC/atsec information security corporation.

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
[EVM]Bound-module-generated RSA public key	RSA public key generated by the bound OpenSSL module	2048, 3072, 4096 bits - 112, 128, 149 bits	Public key - PSP	Key pair generation with RSA		
[EVM]Bound-module-generated RSA private key	RSA private key generated by the bound OpenSSL module	2048, 3072, 4096 bits - 112, 128, 149 bits	Private key - CSP	Key pair generation with RSA		

Table 21: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters	RAM:Plaintext	From service invocation until cipher handle is freed	Zeroized by calling application Module Reset	
ECDSA private key	API input parameters	RAM:Plaintext	From service invocation until cipher handle is freed	Zeroized by calling application Module Reset	
EC Diffie-Hellman private key	API input parameters	RAM:Plaintext	From service invocation until cipher handle is freed	Zeroized by calling application Module Reset	EC Diffie-Hellman public key:Paired With
EC Diffie-Hellman public key	API input parameters	RAM:Plaintext	From service invocation until cipher handle is freed	Zeroized by calling application Module Reset	EC Diffie-Hellman private key:Paired With

© 2026 SUSE LLC/atsec information security corporation.

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

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
EC Diffie-Hellman shared secret	API output parameters	RAM:Plaintext	From service invocation until cipher handle is freed	Zeroized by calling application Module Reset	EC Diffie-Hellman private key:Established By EC Diffie-Hellman public key:Established By
[EVM]Bound-module-generated EC public key	API output parameters	RAM:Plaintext	From service invocation until cipher handle is freed	Zeroized by calling application Automatic Module Reset	[EVM]Bound-module-generated EC private key:Paired With
[EVM]Bound-module-generated EC private key	API output parameters	RAM:Plaintext	From service invocation until cipher handle is freed	Zeroized by calling application Automatic Module Reset	[EVM]Bound-module-generated EC public key:Paired With
[EVM]Bound-module-generated RSA public key	API output parameters	RAM:Plaintext	From service invocation until cipher handle is freed	Zeroized by calling application Automatic Module Reset	[EVM]Bound-module-generated RSA private key:Paired With
[EVM]Bound-module-generated RSA private key	API output parameters	RAM:Plaintext	From service invocation until cipher handle is freed	Zeroized by calling application Automatic Module Reset	[EVM]Bound-module-generated RSA public key:Paired With

Table 22: 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.

© 2026 SUSE LLC/atsec information security corporation.

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	SHA2-256	MAC Tag Verification	SW/FW Integrity	Module becomes operational	[EVM]

Table 23: 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.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A6712) - 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 (A6712) - 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 (A6712) - 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 (A6712) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test

© 2026 SUSE LLC/atsec information security corporation.

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 (A6712) - 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 (A6712) - 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 (A6712) - 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 (A6712) - 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 (A6659) - 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 (A6711) - 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 (A6712) - 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 (A6659) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test

© 2026 SUSE LLC/atsec information security corporation.

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 (A6711) - 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 (A6712) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-XTS Testing Revision 2.0 (A6712) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-XTS Testing Revision 2.0 (A6712) - 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 (A6712) - Encryption	128-bit key, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-CMAC (A6712) - Decryption	128-bit key, decrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A6712)	P-256 curve	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A6712)	P-256 with SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation component	Test runs at power-on before the integrity test

© 2026 SUSE LLC/atsec information security corporation.

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 (A6712)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A6712)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A5883)		KAT	CAST	Module becomes operational	[EVM]Message digest	Test runs at power-on before the integrity test (performed by bound OpenSSL module)
SHA2-256 (A6103)		KAT	CAST	Module becomes operational	[EVM]Message digest	Test runs at power-on before the integrity test (performed by bound OpenSSL module)
SHA2-256 (A6712)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A6712)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A6712)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on

© 2026 SUSE LLC/atsec information security corporation.

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
SHA3-224 (A6712)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A6712)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-384 (A6712)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-512 (A6712)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
Counter DRBG (A5397)	AES with 256-bit key, with/without DF, with/without PR	KAT	CAST	Module becomes operational	[EVM]Instantiate; Generate; Reseed (compliant to SP 800-90A Rev. 1 Section 11.3)	Test runs at power-on before the integrity test
Counter DRBG (A5659)	AES with 256-bit key, with/without DF, with/without PR	KAT	CAST	Module becomes operational	[EVM]Instantiate; Generate; Reseed (compliant to SP 800-90A Rev. 1 Section 11.3)	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A5883)	SHA2-256	KAT	CAST	Module becomes operational	[EVM]Message authentication code	Test runs at power-on before the integrity test

© 2026 SUSE LLC/atsec information security corporation.

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
HMAC-SHA2-256 (A6103)	SHA2-256	KAT	CAST	Module becomes operational	[EVM]Message authentication code	Test runs at power-on before the integrity test
ECDSA KeyGen (FIPS186-5) (A5883)	SHA2-256	PCT	PCT	Key pair generation is successful	[EVM]Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A6103)	SHA2-256	PCT	PCT	Key pair generation is successful	[EVM]Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-5) (A5883)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Key pair generation is successful	[EVM]Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-5) (A6103)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Key pair generation is successful	[EVM]Signature generation and verification	Key pair generation
AES-CFB128 (A6712) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CFB128 (A6712) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CFB8 (A6712) - Encryption	128, 192, 256-bit key, encrypt	KAT	CAST	module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CFB8 (A6712) - Decryption	128, 192, 256-bit key, decrypt	KAT	CAST	module becomes operational	Symmetric operation	Test runs at power-on

© 2026 SUSE LLC/atsec information security corporation.

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

Table 24: 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 Error States).

10.3 Periodic Self-Test Information

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

Table 25: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A6712) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A6712) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A6712) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A6712) - Decryption	KAT	CAST	On Demand	Manually
AES-OFB (A6712) - Encryption	KAT	CAST	On Demand	Manually
AES-OFB (A6712) - Decryption	KAT	CAST	On Demand	Manually
AES-CTR (A6712) - Encryption	KAT	CAST	On Demand	Manually

© 2026 SUSE LLC/atsec information security corporation.

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-CTR (A6712) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A6659) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A6711) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A6712) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A6659) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A6711) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A6712) - Decryption	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A6712) - Encryption	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A6712) - Decryption	KAT	CAST	On Demand	Manually
AES-CMAC (A6712) - Encryption	KAT	CAST	On Demand	Manually
AES-CMAC (A6712) - Decryption	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A6712)	KAT	CAST	On Demand	Manually

© 2026 SUSE LLC/atsec information security corporation.

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
ECDSA SigGen (FIPS186-5) (A6712)	KAT	CAST	On Demand	Manually
SHA-1 (A6712)	KAT	CAST	On Demand	Manually
SHA2-224 (A6712)	KAT	CAST	On Demand	Manually
SHA2-256 (A5883)	KAT	CAST	On Demand	Manually
SHA2-256 (A6103)	KAT	CAST	On Demand	Manually
SHA2-256 (A6712)	KAT	CAST	On Demand	Manually
SHA2-384 (A6712)	KAT	CAST	On Demand	Manually
SHA2-512 (A6712)	KAT	CAST	On Demand	Manually
SHA3-224 (A6712)	KAT	CAST	On Demand	Manually
SHA3-256 (A6712)	KAT	CAST	On Demand	Manually
SHA3-384 (A6712)	KAT	CAST	On Demand	Manually
SHA3-512 (A6712)	KAT	CAST	On Demand	Manually
Counter DRBG (A5397)	KAT	CAST	On Deman	Manually
Counter DRBG (A5659)	KAT	CAST	On Deman	Manually
HMAC-SHA2-256 (A5883)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A6103)	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5883)	PCT	PCT	On Demand	Manually

© 2026 SUSE LLC/atsec information security corporation.

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
ECDSA KeyGen (FIPS186-5) (A6103)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A5883)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A6103)	PCT	PCT	On Demand	Manually
AES-CFB128 (A6712) - Encryption	KAT	CAST	On Demand	Manually
AES-CFB128 (A6712) - Decryption	KAT	CAST	On Demand	Manually
AES-CFB8 (A6712) - Encryption	KAT	CAST	On Demand	Manually
AES-CFB8 (A6712) - Decryption	KAT	CAST	On Demand	Manually

Table 26: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	Error state entered due to self-test failure or failure of the [EVM]	Failure of any CAST Failure of any integrity test Bound OpenSSL module transitions to error state	Unload and reload the module	All cryptographic operations are inhibited. Module does not load or stops functioning.

Table 27: Error States

© 2026 SUSE LLC/atsec information security corporation.

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

When the module fails any pre-operational self-test or conditional test, or when the bound OpenSSL module transitions to any error state, 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.

10.5 Operator Initiation of Self-Tests

On-Demand self-tests can be invoked by calling the ica_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.

© 2026 SUSE LLC/atsec information security corporation.

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

Before the packages specified in Section 11.2 are installed, the SUSE Linux Enterprise SP6 system must operate in the FIPS validated configuration. This can be achieved by:

- Adding 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 FIPS validated configuration after the installation. Execute the fips-mode-setup --enable command. Restart the system.

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

11.2 Administrator Guidance

The module is distributed as part of the following RPM packages:

Version 1.2
libica4-openssl1_1-4.3.1-150600.4.25.1.s390x.rpm

Version 1.3
libica4-4.3.1-150600.4.25.1.s390x.rpm

After the installation of these RPM packages, the Crypto Officer must execute the "Show module name and version" service by checking the outputs of the APIs described below.

For software version, the Crypto officer must check the output of the ica_get_build_version() API. The output of this API must read:

Version 1.2
build: FIPS-SUSE-openssl1_1-4.3.1-150600.4.25.1

Version 1.3
build: FIPS-SUSE--4.3.1-150600.4.25.1

For hardware version, the Crypto officer must check the output of the ica_get_hw_info() API. The output of this API must read:

- vendor_id: IBM/S390
- machine_type: 3931

This identifier corresponds to the IBM z16 A01 hardware with IBM® Telum™ processor.

© 2026 SUSE LLC/atsec information security corporation.

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

11.3 Non-Administrator Guidance

There is no administrator guidance.

11.4 Design and Rules

Not applicable.

11.5 Maintenance Requirements

Not applicable.

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 packages specified in Section11.2 can be uninstalled from the SUSE Linux Enterprise 15 SP6 system.

© 2026 SUSE LLC/atsec information security corporation.

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.

© 2026 SUSE LLC/atsec information security corporation.

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

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
API	Application Programming Interface
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CPACF	Central Processor Assist for Cryptographic Functions
CSP	Critical Security Parameter
CTR	Counter
CVL	Component Validation List
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
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode

© 2026 SUSE LLC/atsec information security corporation.

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

GMAC	Galois Counter Mode Message Authentication Code
HMAC	Keyed-Hash Message Authentication Code
KAS	Key Agreement Scheme
KAT	Known Answer Test
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PAI	Processor Algorithm Implementation
PCT	Pair-wise Consistency Test
PSP	Public Security Parameter
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSP	Sensitive Security Parameter
XOF	Extendable Output Function
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

© 2026 SUSE LLC/atsec information security corporation.

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://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf
FIPS 140-3 IG	**Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program** 18 April 2025 https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf
FIPS 180-4	**Secure Hash Standard (SHS)** August 2015 https://doi.org/10.6028/NIST.FIPS.180-4
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
FIPS 202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 https://doi.org/10.6028/NIST.FIPS.202

© 2026 SUSE LLC/atsec information security corporation.

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

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 800-38D	**Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 https://doi.org/10.6028/NIST.SP.800-38A
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 800-56A 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

© 2026 SUSE LLC/atsec information security corporation.

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

SP 800-133 Rev. 2 **Recommendation for Cryptographic Key Generation**

June 2020

https://doi.org/10.6028/NIST.SP.800-133r2

© 2026 SUSE LLC/atsec information security corporation.

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