SUSE, LLC

SUSE Linux Enterprise Libgcrypt Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Version 1.2

Last update: 2026-07-07

Prepared by:

atsec information security corporation

4516 Seton Center Parkway, Suite 250

Austin, TX 78759

www.atsec.com

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for version 3.2 of the SUSE Linux Enterprise Libgcrypt 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. It has a one-to-one mapping to SP 800-140B starting with section B.2.1 named "General" which maps to Section 1 in this document and ending with section B.2.12 named "Mitigation of other attacks" which maps to Section 12 in this document.

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

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The SUSE Linux Enterprise Libgcrypt Cryptographic Module (hereafter referred to as "the module") is a software library that provides a C language application program interface (API) for use by other applications that require cryptographic functionality. The module operates on a general-purpose computer.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The cryptographic boundary of the module is defined as the libgcrypt shared library and its HMAC file (which stores the expected integrity value for the shared library).

Tested Operational Environment's Physical Perimeter (TOEPP):

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

The cryptographic boundary and TOEPP are schematically represented in Figure 1.

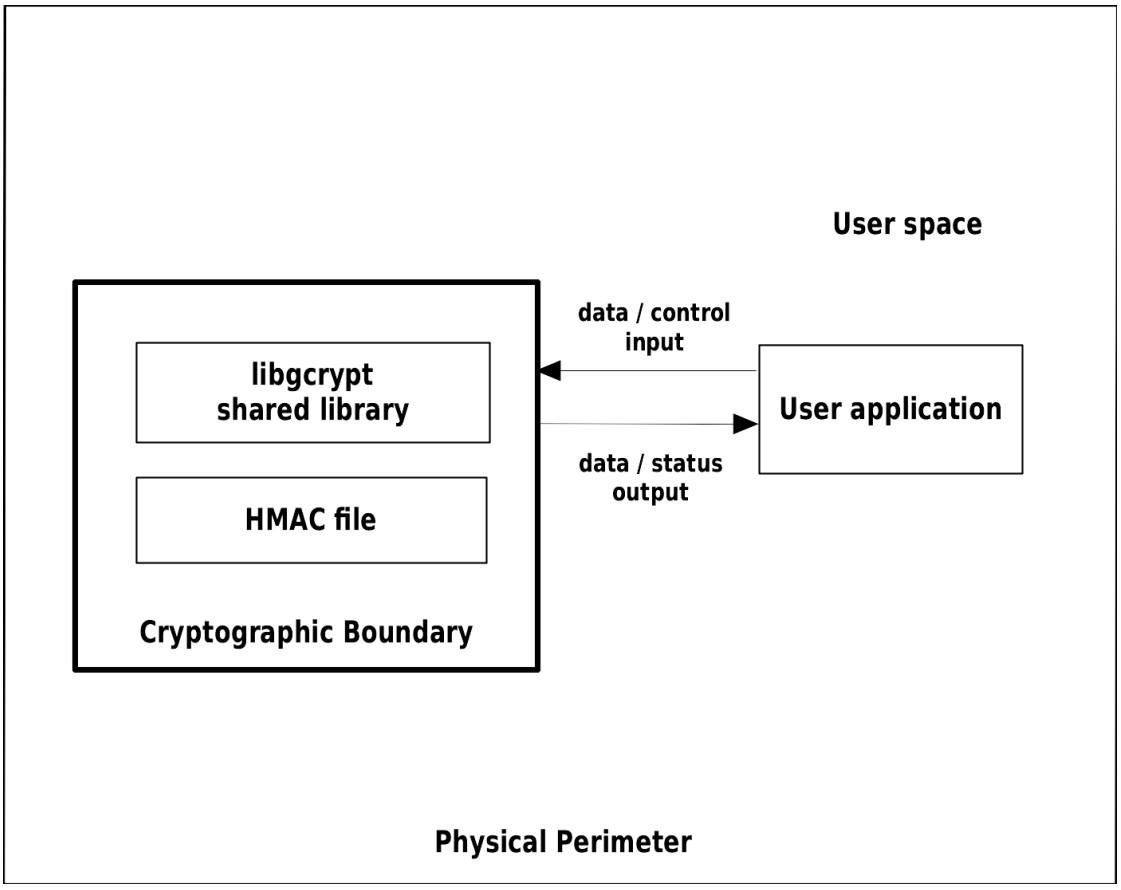

Figure 1: Block Diagram

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
libgcrypt.so.20.3.4 and .libgcrypt.so.hmac on SUSE Linux Enterprise Server 15 SP4 and Intel® Xeon® Silver 4215R or AMD EPYC(TM) 7371	3.2	N/A	HMAC-SHA2-256
libgcrypt.so.20.3.4 and .libgcrypt.so.hmac on SUSE Linux Enterprise Server 15 SP4 and ARM Ampere® Altra® Q80-30	3.2	N/A	HMAC-SHA2-256
libgcrypt.so.20.3.4 and .libgcrypt.so.hmac on SUSE Linux Enterprise Server 15 SP4 and IBM z/15	3.2	N/A	HMAC-SHA2-256
libgcrypt.so.20.3.4 and .libgcrypt.so.hmac on SUSE Linux Enterprise Server 15 SP4 and IBM Power E1080 (9080-HEX)	3.2	N/A	HMAC-SHA2-256

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 SP4	Supermicro Super Server SYS-6019P-WTR	Intel® Xeon® Silver 4215R	Yes	N/A	3.2

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
SUSE Linux Enterprise Server 15 SP4	Supermicro Super Server SYS-6019P-WTR	Intel® Xeon® Silver 4215R	No	N/A	3.2
SUSE Linux Enterprise Server 15 SP4	GIGABYTE R181-Z90-00	AMD EPYC(TM) 7371	Yes	N/A	3.2
SUSE Linux Enterprise Server 15 SP4	GIGABYTE R181-Z90-00	AMD EPYC(TM) 7371	No	N/A	3.2
SUSE Linux Enterprise Server 15 SP4	GIGABYTE G242-P32-QZ	ARM Ampere® Altra® Q80-30	Yes	N/A	3.2
SUSE Linux Enterprise Server 15 SP4	GIGABYTE G242-P32-QZ	ARM Ampere® Altra® Q80-30	No	N/A	3.2
SUSE Linux Enterprise Server 15 SP4	IBM z/15	z15	Yes	N/A	3.2
SUSE Linux Enterprise Server 15 SP4	IBM z/15	z15	No	N/A	3.2
SUSE Linux Enterprise Server 15 SP4 on PowerVM (VIOS 3.1.4.00)	IBM Power E1080 (9080-HEX)	Power10	Yes	N/A	3.2
SUSE Linux Enterprise Server 15 SP4 on PowerVM (VIOS 3.1.4.00)	IBM Power E1080 (9080-HEX)	Power10	No	N/A	3.2

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

The module implements Processor Algorithm Implementation (PAI) for the IBM z/15 platform and Processor Algorithm Acceleration (PAA) for all other tested platforms listed above.

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

Operating System	Hardware Platform
SUSE Linux Enterprise Server 15SP4	IBM LinuxONE III LT1 [z15]
SUSE Linux Enterprise Micro 5.3	Supermicro Super Server SYS-6019P-WTR [Intel® Xeon® Silver 4215R]

Operating System	Hardware Platform
SUSE Linux Enterprise Micro 5.3	GIGABYTE R181-Z90-00 [AMD EPYC(TM) 7371]
SUSE Linux Enterprise Micro 5.3	GIGABYTE G242-P32-QZ [ARM Ampere® Altra® Q80-30]
SUSE Linux Enterprise Micro 5.3	IBM z/15 [z15]
SUSE Linux Enterprise Micro 5.3	IBM LinuxONE III LT1 [z15]
SUSE Linux Enterprise Server for SAP 15SP4	Supermicro Super Server SYS-6019P-WTR [Intel® Xeon® Silver 4215R]
SUSE Linux Enterprise Server for SAP 15SP4	GIGABYTE R181-Z90-00 [AMD EPYC(TM) 7371]
SUSE Linux Enterprise Server for SAP 15SP4	IBM Power E1080 (9080-HEX) [Power10]
SUSE Linux Enterprise Base Container Image 15SP4	Supermicro Super Server SYS-6019P-WTR [Intel® Xeon® Silver 4215R]
SUSE Linux Enterprise Base Container Image 15SP4	GIGABYTE R181-Z90-00 [AMD EPYC(TM) 7371]
SUSE Linux Enterprise Base Container Image 15SP4	GIGABYTE G242-P32-QZ [ARM Ampere® Altra® Q80-30]
SUSE Linux Enterprise Base Container Image 15SP4	IBM z/15 [z15]
SUSE Linux Enterprise Base Container Image 15SP4	IBM LinuxONE III LT1 [z15]
SUSE Linux Enterprise Base Container Image 15SP4	IBM Power E1080 (9080-HEX) [Power10]
SUSE Linux Enterprise Desktop 15SP4	Supermicro Super Server SYS-6019P-WTR [Intel® Xeon® Silver 4215R]
SUSE Linux Enterprise Desktop 15SP4	GIGABYTE R181-Z90-00 [AMD EPYC(TM) 7371]
SUSE Linux Enterprise Real Time 15SP4	Supermicro Super Server SYS-6019P-WTR [Intel® Xeon® Silver 4215R]
SUSE Linux Enterprise Real Time 15SP4	GIGABYTE R181-Z90-00 [AMD EPYC(TM) 7371]

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

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 the allowance FIPS 140-3 Management Manual, 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.

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
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Equivalent to the indicator of the requested service

Table 5: Modes List and Description

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

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

Mode Change Instructions and Status:

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

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3022, A3023, A3025, A3026	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-CCM	A3022, A3023, A3025, A3026	Key Length - 128, 192, 256 Tag Length - 112, 128, 32, 48, 64, 80, 96 IV Length - IV Length: 56, 64, 72, 80, 88, 96, 104 Payload Length - Payload Length: 0-256 Increment 8 AAD Length - AAD Length: 0, 256, 65536	SP 800-38C
AES-CFB128	A3022, A3023, A3025, A3026	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A3022, A3023, A3025, A3026	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3022, A3023, A3025, A3026	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	A3022, A3023, A3025, A3026	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	A3022, A3023, A3025, A3026	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-KW	A3022, A3023, A3025, A3026	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 128-4096 Increment 128	SP 800-38F
AES-OFB	A3022, A3023, A3025, A3026	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A3022, A3023, A3025, A3026	Direction - Decrypt, Encrypt Key Length - 128, 256 Payload Length - Payload Length: 128-65536 Increment 128	SP 800-38E

Algorithm	CAVP Cert	Properties	Reference
		Tweak Mode - Hex Data Unit Length Matches Payload Length - Yes	
Counter DRBG	A3022, A3023, A3025, A3026	Prediction Resistance - No, Yes Supports Reseed - No Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes Additional Input - Additional Input: 0 Entropy Input - Entropy Input: 128, Entropy Input: 192, Entropy Input: 256 Nonce - Nonce: 128, Nonce: 64 Personalization String Length - Personalization String Length: 0 Returned Bits - 1024, 512	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-4)	A3022, A3023, A3024, A3025, A3026	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3022, A3023, A3024, A3025, A3026	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3022, A3023, A3024, A3025, A3026	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3022, A3023, A3024, A3025, A3026	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
Hash DRBG	A3022, A3023, A3024, A3025, A3026	Prediction Resistance - No, Yes Supports Reseed - No Mode - SHA-1, SHA2-256, SHA2-512 Entropy Input - Entropy Input: 160, Entropy Input: 256 Nonce - Nonce: 160, Nonce: 256 Personalization String Length - Personalization String Length: 0, 160, Personalization String Length: 0, 256 Additional Input - Additional Input: 0, 160, Additional Input: 0, 256 Returned Bits - 320, 512, 768	SP 800-90A Rev. 1
HMAC DRBG	A3022, A3023, A3024, A3025, A3026	Prediction Resistance - No, Yes Supports Reseed - No	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
		Mode - SHA-1, SHA2-256, SHA2-512 Entropy Input - Entropy Input: 160, Entropy Input: 256 Nonce - Nonce: 160, Nonce: 256 Personalization String Length - Personalization String Length: 0, 160, Personalization String Length: 0, 256 Additional Input - Additional Input: 0, 160, Additional Input: 0, 256 Returned Bits - 320, 512, 768	
HMAC-SHA-1	A3021, A3022, A3023, A3024, A3025, A3026, A3027	MAC - MAC: 160 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3022, A3023, A3024, A3025, A3026	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3022, A3023, A3024, A3025, A3026	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3022, A3023, A3024, A3025, A3026	MAC - MAC: 384 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3022, A3023, A3024, A3025, A3026	MAC - MAC: 512 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A3024, A3025, A3026	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A3024, A3025, A3026	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A3024, A3025, A3026	MAC - MAC: 384 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A3024, A3025, A3026	MAC - MAC: 512 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
PBKDF	A3022, A3023, A3024, A3025, A3026	Iteration Count - Iteration Count: 10-1000 Increment 1 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Password Length - Password Length: 14-128 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-132
RSA KeyGen (FIPS186-4)	A3022, A3023, A3024, A3025, A3026	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Info Generated By Server - No Public Exponent Mode - Random Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A3022, A3023, A3024, A3025, A3026	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA2-224	FIPS 186-4
RSA SigVer (FIPS186-4)	A3022, A3023, A3024, A3025, A3026	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA2-224 Salt Length - 28 Public Exponent Mode - Fixed Fixed Public Exponent - 010001	FIPS 186-4
SHA-1	A3021, A3022, A3023, A3024, A3025, A3026, A3027	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-224	A3022, A3023, A3024, A3025, A3026	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A3022, A3023, A3024, A3025, A3026	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A3022, A3023, A3024, A3025, A3026	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-512	A3022, A3023, A3024, A3025, A3026	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA3-224	A3024, A3025, A3026	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-256	A3024, A3025, A3026	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-384	A3024, A3025, A3026	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-512	A3024, A3025, A3026	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHAKE-128	A3024, A3025, A3026	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	A3024, A3025, A3026	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 6: Approved Algorithms

Vendor-Affirmed Algorithms:

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

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

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

Non-Approved, Allowed Algorithms with No Security Claimed:

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

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES EAX	Encryption; Decryption
AES OCB	Encryption; Decryption
AES GCM	Authenticated encryption; Authenticated decryption
AES GMAC	Message authentication code (MAC)
CRC32	Error detection code
BLAKE2B-160, BLAKE2B-256, BLAKE2B-384, BLAKE2B-512, BLAKE2S-128, BLAKE2S-160, BLAKE2S-224, BLAKE2S-256	Message digest
GOST R 34.11	Message digest
MD4, MD5	Message digest
RIPEMD-160	Message digest
Tiger	Message digest
SM3, STRIBOG-256, STRIBOG-512	Message digest
Whirlpool	Message digest
PBKDF2 with non-approved message digest algorithms or using input parameters not meeting requirements stated in Security Policy, section 2.7	Key derivation
RSA OAEP	Key encapsulation
ElGamal	Key pair generation

Table 8: Non-Approved, Not Allowed Algorithms

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

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric encryption	BC-UnAuth	Encrypt a plaintext		AES-CBC: (A3022, A3023, A3025, A3026) AES-CFB128: (A3022, A3023, A3026) AES-CFB8: (A3022, A3023, A3025, A3026) AES-CTR: (A3022, A3023, A3025, A3026) AES-ECB: (A3022, A3023, A3025, A3026) AES-OFB: (A3022, A3023, A3025, A3026) AES-XTS Testing Revision 2.0: (A3022, A3023, A3025, A3026)
Symmetric decryption	BC-UnAuth	Decrypt a ciphertext		AES-CBC: (A3022, A3023, A3025, A3026) AES-CFB128: (A3022, A3023, A3025, A3026) AES-CFB8: (A3022, A3023, A3025, A3026) AES-CTR: (A3022, A3023, A3025, A3026) AES-ECB: (A3022, A3023, A3025, A3026) AES-OFB: (A3022, A3023, A3025, A3026) AES-XTS Testing Revision 2.0:

Name	Type	Description	Properties	Algorithms
				(A3022, A3023, A3025, A3026)
Authenticated symmetric encryption	BC-Auth	Encrypt and generate a MAC for a plaintext		AES-CCM: (A3022, A3023, A3025, A3026)
Authenticated symmetric decryption	BC-Auth	Verify and decrypt a plaintext		AES-CCM: (A3022, A3023, A3025, A3026)
Key wrapping	BC-Auth	Key wrapping (as a standalone service)		AES-KW: (A3022, A3023, A3025, A3026)
Key unwrapping	BC-Auth	Key unwrapping (as a standalone service)		AES-KW: (A3022, A3023, A3025, A3026)
Message authentication	MAC	Compute a MAC tag		AES-CMAC: (A3022, A3023, A3025, A3026) HMAC-SHA-1: (A3021, A3022, A3023, A3024, A3025, A3026, A3027) HMAC-SHA2-224: (A3022, A3023, A3024, A3025, A3026) HMAC-SHA2-256: (A3022, A3023, A3024, A3025, A3026) HMAC-SHA2-384: (A3022, A3023, A3024, A3025, A3026) HMAC-SHA2-512: (A3022, A3023, A3024, A3025, A3026) HMAC-SHA3-224: (A3024, A3025, A3026)

Name	Type	Description	Properties	Algorithms
				HMAC-SHA3-256: (A3024, A3025, A3026) HMAC-SHA3-384: (A3024, A3025, A3026) HMAC-SHA3-512: (A3024, A3025, A3026)
Random number generation	DRBG	Generate random bytes		Counter DRBG: (A3022, A3023, A3025, A3026) Hash DRBG: (A3022, A3023, A3024, A3025, A3026) HMAC DRBG: (A3022, A3023, A3024, A3025, A3026)
Key pair generation	AsymKeyPair-KeyGen CKG	Generate an asymmetric key pair		ECDSA KeyGen (FIPS186-4): (A3022, A3023, A3024, A3025, A3026) RSA KeyGen (FIPS186-4): (A3022, A3023, A3024, A3025, A3026) CKG: ()
Key validation	AsymKeyPair-KeyVer	Validate an asymmetric key pair		ECDSA KeyVer (FIPS186-4): (A3022, A3023, A3024, A3025, A3026)
Digital signature generation	DigSig-SigGen	Generate a digital signature on a message		ECDSA SigGen (FIPS186-4): (A3022, A3023, A3024, A3025, A3026) RSA SigGen

Name	Type	Description	Properties	Algorithms
				(FIPS186-4): (A3022, A3023, A3024, A3025, A3026)
Digital signature verification	DigSig-SigVer	Verify a digital signature on a message		ECDSA SigVer (FIPS186-4): (A3022, A3023, A3024, A3025, A3026) RSA SigVer (FIPS186-4): (A3022, A3023, A3024, A3025, A3026)
Key derivation	PBKDF	Derive keying material from a password		PBKDF: (A3022, A3023, A3024, A3025, A3026)
Message digest	SHA XOF	Compute a message digest		SHA-1: (A3021, A3022, A3023, A3024, A3025, A3026, A3027) SHA2-224: (A3022, A3023, A3024, A3025, A3026) SHA2-256: (A3022, A3023, A3024, A3025, A3026) SHA2-384: (A3022, A3023, A3024, A3025, A3026) SHA2-512: (A3022, A3023, A3024, A3025, A3026) SHA3-224: (A3024, A3025, A3026) SHA3-256: (A3024, A3025, A3026) SHA3-384: (A3024, A3025, A3026) SHA3-512: (A3024, A3025, A3026) SHAKE-128: (A3024, A3025,

Name	Type	Description	Properties	Algorithms
				A3026) SHAKE-256: (A3024, A3025, A3026)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES XTS

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

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 are not identical.

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 RSA

In compliance with IG C.F, the module implements only the approved modulus sizes of 2048, 3072, and 4096 bits for signature generation and verification. Each algorithm was tested with all key sizes. The corresponding certificates can be found in Section 2.5.

2.7.3 Legacy Use

Digital signature verification using SHA-1 is allowed for legacy use only.

This legacy algorithm can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M.

2.7.4 Key Derivation using SP 800-132 PBKDF

The module provides password-based key derivation (PBKDF), compliant with SP 800-132. The module supports option 1a from Section 5.4 of SP 800-132, in which the Master Key (MK) or a segment of it is used directly as the Data Protection Key (DPK).

In accordance with SP 800-132 and FIPS 140-3 IG D.N, the following requirements are met:

- Derived keys shall be used only for storage applications and shall not be used for any other purposes. The length of the MK or DPK is 112 bits or more.
- Passwords or passphrases, used as an input for PBKDF2, shall not be used as cryptographic keys.
- The minimum length of the password or passphrase accepted by the module is 20 characters. The probability of guessing the value, assuming a worst-case scenario of all digits, is estimated to be at most

10^{-20}. Combined with the minimum iteration count as described below, this provides an acceptable trade-off between user experience and security against brute-force attacks.

- A portion of the salt shall be generated randomly using the SP 800-90A Rev. 1 DRBG provided by the module. The minimum length required is 128 bits.
- The iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The minimum value accepted by the module is 1000.

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

2.7.5 Key Transport

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

2.8 RBG and Entropy

Cert Number	Vendor Name
E31	SUSE LLC

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Libgcrypt CPU Time Jitter RNG	Non-Physical	SUSE Linux Enterprise Server 15 SP4 on Intel® Xeon® Silver 4215R, SUSE Linux Enterprise Server 15 SP4 on AMD EPYC(TM) 7371, SUSE Linux Enterprise Server 15 SP4 on ARM Ampere® Altra® Q80-30, SUSE Linux Enterprise Server 15 SP4 on z15, SUSE Linux Enterprise Server 15 SP4 on Power10	256 bits	Full entropy	SHA3-256 (A3035)

Table 11: Entropy Sources

The module implements three different Deterministic Random Bit Generator (DRBG) implementations based on SP 800-90A Rev. 1: CTR_DRBG, Hash_DRBG, and HMAC_DRBG. Each of these DRBG implementations can be instantiated by the operator of the module. When instantiated, these DRBGs can be used to generate random numbers for external usage.

Additionally, the module employs a specific HMAC-SHA-512 DRBG implementation for internal purposes (e.g. to generate asymmetric key pairs). This DRBG is initially seeded with 384 output bits from the entropy source (corresponding to 384 bits of entropy) and reseeded with 256 output bits from the entropy source

(corresponding to 256 bits of entropy). A different DRBG mechanism may be selected by invoking the gcry_control(GCRYCTL_DRBG_REINIT) function.

The module complies with the Public Use Document for ESV certificate E31 by seeding the aforementioned DRBG using the jent_read_entropy() function, which corresponds to the GetEntropy() function. The operational environment of the module is identical to the one listed on the ESV certificate. There are no maintenance requirements for the entropy source.

2.9 Key Generation

The module implements asymmetric key pair generation compliant with SP 800-133 Rev. 2. When random values are required, they are directly obtained as output from an SP 800-90A Rev. 1 approved DRBG, compliant with Section 4 of SP 800-133 Rev. 2 (without XOR). The following methods are implemented:

- RSA key pair generation
- ECDSA key pair generation

Details on these implementations can be found on the corresponding certificates listed in the Approved Algorithms table.

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

Additionally, the module provides password-based key derivation (PBKDF) compliant with SP 800-133 Rev.2, Section 6.2, which maps to SP 800-132.

2.10 Key Establishment

The module does not provide any key establishment services. However, it does provide authenticated encryption algorithms which can be used by external operators/applications as part of an approved key transport scheme.

2.11 Industry Protocols

The module does not implement any industry protocols.

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.
N/A	Status Output	API return codes, API output parameters for status output.

Table 12: Ports and Interfaces

As a software-only module, the module does not have physical ports. The operator can only interact with the module through the API provided by the module. Thus, the physical ports are interpreted to be the physical ports of the hardware platform on which the module runs.

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. All data output via the data output interface is inhibited when the module is performing pre-operational self-tests, zeroization, or when the module is in an error state.

The module does not implement a control output interface.

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 13: Roles

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

4.3 Approved Services

The table below lists the approved services provided by the module. 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:

- **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.

The details of the approved cryptographic algorithms including the CAVP certificate numbers can be found in Section 2.5.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric Encryption	Encrypt a plaintext	_gcry_fips_indicator_cipher function returns "0"	Plaintext, key, IV	Ciphertext	Symmetric encryption	Crypto Officer - AES key: W,E
Symmetric Decryption	Decrypt a ciphertext	_gcry_fips_indicator_cipher function returns "0"	Ciphertext, key, IV	Plaintext	Symmetric decryption	Crypto Officer - AES key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Authenticated Symmetric Encryption	Encrypt a plaintext and generate a MAC	_gcry_fips_indicator_cipher function returns "0"	Plaintext, key, IV	Ciphertext, MAC tag	Authenticated symmetric encryption	Crypto Officer - AES key: W,E
Authenticated Symmetric Decryption	Verify and decrypt a ciphertext	_gcry_fips_indicator_cipher function returns "0"	Ciphertext, MAC, key, IV	Plaintext or Fail	Authenticated symmetric decryption	Crypto Officer - AES key: W,E
Key wrapping	Perform AES-based key wrapping	_gcry_fips_indicator_cipher function returns "0"	AES key, key to be wrapped	Wrapped key	Key wrapping	Crypto Officer - AES key: W,E
Key unwrapping	Perform AES-based key unwrapping	_gcry_fips_indicator_cipher function returns "0"	Wrapped key, AES key	Unwrapped key	Key unwrapping	Crypto Officer - AES key: W,E
Key pair generation	Generate a key pair	_gcry_fips_indicator_pk (and rsa_generate if RSA key pair) returns "0"	Key size	Key pair	Key pair generation	Crypto Officer - Module-generated RSA public key: G,R - Module-generated RSA private key: G,R - Module-generated ECDSA public key: G,R - Module-generated ECDSA private key: G,R - Intermedia

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						te key generation value: G,E,Z
Public key verification	Verify a public key	_gcry_fips_indicator_pk function returns "0"	Key	Return codes/log messages	Key validation	Crypto Officer - ECDSA public key: W,E
Digital signature generation	Generate a signature	_gcry_fips_indicator_pk (and rsa_sign if RSA) returns "0"	Private key, message, hash algorithm	Signature	Digital signature generation	Crypto Officer - RSA private key: W,E - ECDSA private key: W,E
Digital signature verification	Verify a signature	_gcry_fips_indicator_pk (and rsa_verify if RSA) returns "0"	Signature, message, hash algorithm, public key	Signature verification result	Digital signature verification	Crypto Officer - RSA public key: W,E - ECDSA public key: W,E
Message authentication	Compute a MAC tag	_gcry_fips_indicator_mac function returns "0"	Message, AES key or HMAC key	MAC tag	Message authentication	Crypto Officer - AES key: W,E - HMAC key: W,E
Message digest	Compute a message digest	_gcry_fips_indicator_hash function returns "0"	Message	Message digest	Message digest	Crypto Officer
Random number generation	Generate random bitstrings	drbg_generate function returns "0"	Size	Random number	Random number generation	Crypto Officer - Entropy input: W,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- DRBG seed: G,E,Z - DRBG Internal state (V, Key): G,W,E - DRBG Internal state (V, C): G,W,E
XOF	Compute the output of an XOF	_gcry_fips_indicator_hash function returns "0"	Message	Message digest	Message digest	Crypto Officer
Key derivation from a password	Derive a key from a password	_gcry_fips_indicator_kdf function returns "0"	Password or passphrase	Derived key	Key derivation	Crypto Officer - Password or passphrase: W,E - Derived key: G,R
On demand integrity test	Perform the integrity test on demand	None	N/A	Pass/Fail	Message authentication	Crypto Officer
Show status	Return the module status	None	N/A	Module status	None	Crypto Officer
Zeroization	Zeroize all SSPs	None	Any SSP	N/A	None	Crypto Officer - AES key: Z - HMAC key: Z - RSA public key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- RSA private key: Z - ECDSA public key: Z - ECDSA private key: Z - Intermediate key generation value: Z - Password or passphrase: Z - Derived key: Z - Entropy input: Z - DRBG seed: Z - DRBG Internal state (V, Key): Z - DRBG Internal state (V, C): Z
Self-tests	Perform self-tests	None	N/A	Pass/Fail	Symmetric encryption Symmetric decryption Authenticated symmetric encryption Authenticated symmetric decryption	Crypto Officer - AES key: E - HMAC key: E - RSA public key: E - RSA private key: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Message authentication Random number generation Digital signature generation Digital signature verification Key derivation Message digest	- ECDSA public key: E - ECDSA private key: E - Intermediate key generation value: E - Password or passphrase: E - Entropy input: E
Show version	Return the name and version information	None	N/A	Name and version information	None	Crypto Officer

Table 14: Approved Services

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Encryption	AES encryption using non-approved AES modes	AES EAX AES OCB	CO
Decryption	AES decryption using non-approved AES modes	AES EAX AES OCB	CO
Authenticated encryption	Authenticated AES encryption using non-approved AES modes	AES GCM	CO
Authenticated decryption	Authenticated AES decryption using non-approved AES modes	AES GCM	CO

Name	Description	Algorithms	Role
Message authentication	Generate a message authentication code (MAC)	AES GMAC	CO
Message digest	Message digest using non-approved algorithms	BLAKE2B-160, BLAKE2B-256, BLAKE2B-384, BLAKE2B-512, BLAKE2S-128, BLAKE2S-160, BLAKE2S-224, BLAKE2S-256 GOST R 34.11 MD4, MD5 RIPEMD-160 Tiger SM3, STRIBOG-256, STRIBOG-512 Whirlpool	CO
Error detection code	Error detection code	CRC32	CO
Key generation	Generate a key pair	ElGamal	CO
Key derivation	PBKDF2 Key derivation	PBKDF2 with non-approved message digest algorithms or using input parameters not meeting requirements stated in Security Policy, section 2.7	CO
Key encapsulation	Encapsulate a key	RSA OAEP	CO

Table 15: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not load external software or firmware.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified by comparing an HMAC-SHA-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 MAC key is hardcoded within the module.

5.2 Initiate on Demand

Integrity tests are performed as part of the pre-operational self-tests, which are executed when the module is initialized. The integrity tests can be invoked on demand by invoking the gry_control(GCRYCTL_SELFTEST) API function call or by unloading and subsequently re-initializing the module (i.e. rebooting the system), which will perform (among others) the software integrity tests.

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 11.1.

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.

7 Physical Security

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

8 Non-Invasive Security

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

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.	Dynamic

Table 16: Storage Areas

The module does not perform persistent storage of SSPs. The SSPs are temporarily stored in the RAM in plaintext form. SSPs are stored until they are zeroized by the operator (using a zeroization call or removing power from the module) or zeroized automatically.

9.2 SSP Input-Output Methods

The module does not support 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 in 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 17: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Wipe and Free memory block allocated	Zeroizes the SSPs contained within the cipher handle.	Memory occupied by SSPs is overwritten with zeroes and then it is released, which renders the SSP values irretrievable.	By calling gcry_free(), which also invokes any additional zeroization functions including: AES key: gcry_cipher_close(); HMAC key: gcry_mac_close(); RSA public and private key: gcry_sexp_release(), gcry_mpi_release(); ECDSA public and private key: gcry_sexp_release(), gcry_mpi_release(), gcry_mpi_point_release();

Zeroization Method	Description	Rationale	Operator Initiation
		The completion of the zeroization routine indicates that the zeroization procedure succeeded.	DRBG internal state and Entropy input: gcry_ctrl(GCRYCTL_TERM_SECMEM)
Automatic	Automatically zeroized by the module when no longer needed	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable.	N/A
Module Reset	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed.	By unloading and reloading the module

Table 18: SSP Zeroization Methods

The memory occupied by SSPs is allocated by regular memory allocation operating system calls. The application acting as the CO is responsible for calling the appropriate zeroization functions provided in the module's API and listed in Section 4.3. The zeroization functions overwrite the memory occupied by SSPs with zeros then deallocate the memory using memory deallocation operating system calls. The completion of a zeroization routine serves as an indicator that zeroization has succeeded.

9.4 SSPs

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			Symmetric encryption Symmetric decryption Authenticated symmetric encryption Authenticated symmetric decryption Message

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						authentication Key wrapping Key unwrapping
HMAC key	HMAC key	112-524288 bits - 112-256 bits	Symmetric key - CSP			Message authentication
Module-generated RSA public key	RSA public key generated by the module	2048, 3072, 4096 bits - 112, 128, 149 bits	Public key - PSP	Key pair generation		
Module-generated RSA private key	RSA private key generated by the module	2048, 3072, 4096 bits - 112, 128, 149 bits	Private key - CSP	Key pair generation		
RSA public key	RSA public key written to the module	2048, 3072, 4096 bits - 112, 128, 149 bits	Public key - PSP			Digital signature verification
RSA private key	RSA private key written to the module	2048, 3072, 4096 bits - 112, 128, 149 bits	Private key - CSP			Digital signature generation
Module-generated ECDSA public key	ECDSA public key generated by the module	P-256, P-384, P-521 bits - 128, 192, 256 bits	Public key - CSP	Key pair generation		
Module-generated ECDSA private key	ECDSA private key generated by the module	P-256, P-384, P-521 bits - 128, 192, 256 bits	Private key - CSP	Key pair generation		
ECDSA public key	ECDSA public key written to the module	P-256, P-384, P-521 bits - 128, 192, 256 bits	Public key - CSP			Digital signature verification

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ECDSA private key	ECDSA private key written to the module	P-256, P-384, P-521 bits - 128, 192, 256 bits	Private key - CSP			Digital signature verification
Intermediate key generation value	Intermediate key generation value generated during key generation and key derivation services	112-4096 bits - 112-256 bits	Intermediate value - CSP	Key pair generation		Key pair generation
Password or passphrase	Password or passphrase	20-128 characters - N/A	Password - CSP			Key derivation
Derived key	Derived key	112-256 bits - 112-256 bits	Symmetric key - CSP	Key derivation		
Entropy input	Entropy input used to seed DRBGs	128-384 bits - 128-384 bits	Entropy Input - CSP	Random number generation		Random number generation
DRBG seed	DRBG seed derived from entropy input and additional data	CTR_DRBG: 256, 320, 384 bits Hash_DBRG: 440, 888 bits HMAC_DRBG: 440, 888 bits - CTR_DRBG: 128, 192, 256 bits Hash_DBRG: 128, 256 bits HMAC_DRBG: 128, 256 bits	Seed - CSP	Random number generation		Random number generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG Internal state (V, Key)	Internal state of CTR_DRBG and HMAC_DRBG	CTR_DRBG: 256, 320, 384 bits; HMAC_DRBG: 320, 512, 1024 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256 bits	Internal state - CSP	Random number generation		Random number generation
DRBG Internal state (V, C)	Internal state of Hash_DRBG	880, 1776 bits - 128, 256 bits	Internal state - CSP	Random number generation		Random number generation

Table 19: 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	
HMAC key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	
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 Intermediate key generation value:Derived from
Module-generated RSA private key	API output parameters	RAM:Plaintext	From service invocation to	Wipe and Free memory block	RSA public key:Paired with Intermediate key

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			service completion	allocated Module Reset	generation value:Derived from
RSA public key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	RSA private key:Paired with
RSA private key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	RSA public key:Paired with
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 Intermediate key generation value:Derived from
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 Intermediate key generation value:Derived from
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
Intermediate key generation value		RAM:Plaintext	From service invocation to service completion	Automatic	Module-generated ECDSA public key:Derives Module-generated

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					ECDSA private key:Derives Module-generated RSA public key:Derives Module-generated RSA private key:Derives
Password or passphrase	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Derived key:Derives
Derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Password or Passphrase:Derived from
Entropy input		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DRBG seed:Derives
DRBG seed		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Entropy input:Derived from DRBG internal state (V, Key):Derives DRBG internal state (V, C):Derives
DRBG Internal state (V, Key)		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DRBG seed:Derived from
DRBG Internal state (V, C)		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DRBG seed:Derived from

Table 20: 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.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A3022)	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A3023)	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A3024)	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A3025)	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A3026)	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	N/A

Table 21: Pre-Operational Self-Tests

The pre-operational software integrity tests are performed automatically when the module is powered on, before the module transitions into the operational state. The module transitions to the operational state only after the pre-operational self-tests are passed successfully. If any pre-operational self-test fails, the module immediately transitions to the error state.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A3025) - Encryption	128, 192, and 256-bit keys; Encrypt	KAT	CAST	Module becomes operational	Symmetric encryption	Test runs at power-on before the integrity test
AES-ECB (A3023) - Encryption	128, 192, and 256-bit keys; Encrypt	KAT	CAST	Module becomes operational	Symmetric encryption	Test runs at power-on before the integrity test
AES-ECB (A3022) - Encryption	128, 192, and 256-bit keys; Encrypt	KAT	CAST	Module becomes operational	Symmetric encryption	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
AES-ECB (A3026) - Encryption	128, 192, and 256-bit keys; Encrypt	KAT	CAST	Module becomes operational	Symmetric encryption	Test runs at power-on before the integrity test
AES-ECB (A3025) - Decryption	128, 192, and 256-bit keys; Decrypt	KAT	CAST	Module becomes operational	Symmetric decryption	Test runs at power-on before the integrity test
AES-ECB (A3023) - Decryption	128, 192, and 256-bit keys; Decrypt	KAT	CAST	Module becomes operational	Symmetric decryption	Test runs at power-on before the integrity test
AES-ECB (A3022) - Decryption	128, 192, and 256-bit keys; Decrypt	KAT	CAST	Module becomes operational	Symmetric decryption	Test runs at power-on before the integrity test
AES-ECB (A3026) - Decryption	128, 192, and 256-bit keys; Decrypt	KAT	CAST	Module becomes operational	Symmetric decryption	Test runs at power-on before the integrity test
AES-CMAC (A3025)	128 bit-key	KAT	CAST	Module becomes operational	generate	Test runs at power-on before the integrity test
AES-CMAC (A3023)	128 bit-key	KAT	CAST	Module becomes operational	generate	Test runs at power-on before the integrity test
AES-CMAC (A3022)	128 bit-key	KAT	CAST	Module becomes operational	generate	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CMAC (A3026)	128 bit-key	KAT	CAST	Module becomes operational	generate	Test runs at power-on before the integrity test
Counter DRBG (A3025)	128-bit key	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
Counter DRBG (A3023)	128-bit key	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
Counter DRBG (A3022)	128-bit key	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
Counter DRBG (A3026)	128-bit key	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
Hash DRBG (A3025)	SHA-1, SHA-256, and SHA-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
Hash DRBG (A3023)	SHA-1, SHA-256, and SHA-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
Hash DRBG (A3022)	SHA-1, SHA-256, and SHA-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
Hash DRBG (A3024)	SHA-1, SHA-256, and SHA-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Hash DRBG (A3026)	SHA-1, SHA-256, and SHA-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
HMAC DRBG (A3025)	HMAC-SHA-1, HMAC-SHA2-256, and HMAC-SHA2-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
HMAC DRBG (A3023)	HMAC-SHA-1, HMAC-SHA2-256, and HMAC-SHA2-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
HMAC DRBG (A3022)	HMAC-SHA-1, HMAC-SHA2-256, and HMAC-SHA2-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
HMAC DRBG (A3024)	HMAC-SHA-1, HMAC-SHA2-256, and HMAC-SHA2-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
HMAC DRBG (A3026)	HMAC-SHA-1, HMAC-SHA2-256, and HMAC-SHA2-512 with/without prediction resistance	KAT	CAST	Module becomes operational	Random number generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3025)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3023)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigGen (FIPS186-4) (A3022)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3024)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3026)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3025)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3023)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3022)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3024)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3026)	P-256 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3027)	SHA-1	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA-1 (A3025)	SHA-1	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3023)	SHA-1	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3021)	SHA-1	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3022)	SHA-1	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3024)	SHA-1	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3026)	SHA-1	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3025)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3023)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3022)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-224 (A3024)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3026)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3025)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3023)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3022)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3024)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3026)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3025)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3023)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-384 (A3022)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3024)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3026)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3025)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3023)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3022)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3024)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3026)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-224 (A3025)	SHA3-224	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA3-224 (A3024)	SHA3-224	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-224 (A3026)	SHA3-224	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-256 (A3025)	SHA3-256	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-256 (A3024)	SHA3-256	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-256 (A3026)	SHA3-256	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-384 (A3025)	SHA3-384	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-384 (A3024)	SHA3-384	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-384 (A3026)	SHA3-384	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-512 (A3025)	SHA3-512	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA3-512 (A3024)	SHA3-512	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
HMAC-SHA3-512 (A3026)	SHA3-512	KAT	CAST	Module becomes operational	Message authentication code computation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A3025)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A3023)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A3022)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A3024)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A3026)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A3025)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A3023)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigVer (FIPS186-4) (A3022)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A3024)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A3026)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
SHA-1 (A3027)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3025)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3023)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3021)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3022)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3024)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (A3026)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3025)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3023)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3022)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3024)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3026)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3025)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3023)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3022)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-256 (A3024)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3026)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3025)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3023)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3022)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3024)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3026)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3025)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3023)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-512 (A3022)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3024)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3026)	0-65536-bit messages	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
PBKDF (A3025)	SHA-1, SHA-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3023)	SHA-1, SHA-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3022)	SHA-1, SHA-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3024)	SHA-1, SHA-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3026)	SHA-1, SHA-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
ECDSA KeyGen (FIPS186-4) (A3025)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA KeyGen (FIPS186-4) (A3023)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A3022)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A3024)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A3026)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A3025)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A3023)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A3022)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A3024)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A3026)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Successful key pair generation	Signature generation and verification	Key pair generation

Table 22: Conditional Self-Tests

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

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A3022)	MAC tag verification	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-256 (A3023)	MAC tag verification	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-256 (A3024)	MAC tag verification	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-256 (A3025)	MAC tag verification	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-256 (A3026)	MAC tag verification	SW/FW Integrity	On Demand	Manually

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A3025) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A3023) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A3022) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A3026) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A3025) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A3023) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A3022) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A3026) - Decryption	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CMAC (A3025)	KAT	CAST	On Demand	Manually
AES-CMAC (A3023)	KAT	CAST	On Demand	Manually
AES-CMAC (A3022)	KAT	CAST	On Demand	Manually
AES-CMAC (A3026)	KAT	CAST	On Demand	Manually
Counter DRBG (A3025)	KAT	CAST	On Demand	Manually
Counter DRBG (A3023)	KAT	CAST	On Demand	Manually
Counter DRBG (A3022)	KAT	CAST	On Demand	Manually
Counter DRBG (A3026)	KAT	CAST	On Demand	Manually
Hash DRBG (A3025)	KAT	CAST	On Demand	Manually
Hash DRBG (A3023)	KAT	CAST	On Demand	Manually
Hash DRBG (A3022)	KAT	CAST	On Demand	Manually
Hash DRBG (A3024)	KAT	CAST	On Demand	Manually
Hash DRBG (A3026)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3025)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3023)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC DRBG (A3022)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3024)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3026)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3025)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3023)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3022)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3024)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3026)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3025)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3023)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3022)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3024)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigVer (FIPS186-4) (A3026)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3027)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3025)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3023)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3021)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3022)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3024)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3026)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3025)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3023)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3022)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3024)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3026)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3025)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3023)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A3022)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3024)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3026)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3025)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3023)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3022)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3024)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3026)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3025)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3023)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3022)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3024)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3026)	KAT	CAST	On Demand	Manually
HMAC-SHA3-224 (A3025)	KAT	CAST	On Demand	Manually
HMAC-SHA3-224 (A3024)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA3-224 (A3026)	KAT	CAST	On Demand	Manually
HMAC-SHA3-256 (A3025)	KAT	CAST	On Demand	Manually
HMAC-SHA3-256 (A3024)	KAT	CAST	On Demand	Manually
HMAC-SHA3-256 (A3026)	KAT	CAST	On Demand	Manually
HMAC-SHA3-384 (A3025)	KAT	CAST	On Demand	Manually
HMAC-SHA3-384 (A3024)	KAT	CAST	On Demand	Manually
HMAC-SHA3-384 (A3026)	KAT	CAST	On Demand	Manually
HMAC-SHA3-512 (A3025)	KAT	CAST	On Demand	Manually
HMAC-SHA3-512 (A3024)	KAT	CAST	On Demand	Manually
HMAC-SHA3-512 (A3026)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3025)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3023)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3022)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3024)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigGen (FIPS186-4) (A3026)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3025)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3023)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3022)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3024)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3026)	KAT	CAST	On Demand	Manually
SHA-1 (A3027)	KAT	CAST	On Demand	Manually
SHA-1 (A3025)	KAT	CAST	On Demand	Manually
SHA-1 (A3023)	KAT	CAST	On Demand	Manually
SHA-1 (A3021)	KAT	CAST	On Demand	Manually
SHA-1 (A3022)	KAT	CAST	On Demand	Manually
SHA-1 (A3024)	KAT	CAST	On Demand	Manually
SHA-1 (A3026)	KAT	CAST	On Demand	Manually
SHA2-224 (A3025)	KAT	CAST	On Demand	Manually
SHA2-224 (A3023)	KAT	CAST	On Demand	Manually
SHA2-224 (A3022)	KAT	CAST	On Demand	Manually
SHA2-224 (A3024)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-224 (A3026)	KAT	CAST	On Demand	Manually
SHA2-256 (A3025)	KAT	CAST	On Demand	Manually
SHA2-256 (A3023)	KAT	CAST	On Demand	Manually
SHA2-256 (A3022)	KAT	CAST	On Demand	Manually
SHA2-256 (A3024)	KAT	CAST	On Demand	Manually
SHA2-256 (A3026)	KAT	CAST	On Demand	Manually
SHA2-384 (A3025)	KAT	CAST	On Demand	Manually
SHA2-384 (A3023)	KAT	CAST	On Demand	Manually
SHA2-384 (A3022)	KAT	CAST	On Demand	Manually
SHA2-384 (A3024)	KAT	CAST	On Demand	Manually
SHA2-384 (A3026)	KAT	CAST	On Demand	Manually
SHA2-512 (A3025)	KAT	CAST	On Demand	Manually
SHA2-512 (A3023)	KAT	CAST	On Demand	Manually
SHA2-512 (A3022)	KAT	CAST	On Demand	Manually
SHA2-512 (A3024)	KAT	CAST	On Demand	Manually
SHA2-512 (A3026)	KAT	CAST	On Demand	Manually
PBKDF (A3025)	KAT	CAST	On Demand	Manually
PBKDF (A3023)	KAT	CAST	On Demand	Manually
PBKDF (A3022)	KAT	CAST	On Demand	Manually
PBKDF (A3024)	KAT	CAST	On Demand	Manually
PBKDF (A3026)	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3025)	PCT	PCT	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA KeyGen (FIPS186-4) (A3023)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3022)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3024)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3026)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3025)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3023)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3022)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3024)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3026)	PCT	PCT	On Demand	Manually

Table 24: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Self-test error state	Enters an error state in which the module state can be obtained.	Failure of pre-operational tests or conditional tests.	Reboot the module	An error message related to the cause of the failure, gcry_control(GCRYCTL_OPERATIONAL_P) returns FALSE.
Abort error state	The module is aborted and the module state cannot be obtained.	Random numbers are requested in the error state or cipher operations are requested on a deallocated handle.	Reboot the module	The module is aborted and is not available for use.

Table 25: Error States

In the error state, the output interface is inhibited, and the module accepts no more inputs or requests (as the module is no longer running).

10.5 Operator Initiation of Self-Tests

The pre-operational and conditional known-answer self-tests can be executed on-demand by unloading and subsequently re-initializing the module. The pair-wise consistency tests can be invoked on demand by requesting a key-pair generation service.

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 using the zypper tool as follows:

zypper install libgcrypt20

zypper install libgcrypt20-hmac

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 operating environment needs to be configured to support FIPS, 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.

Now, the operating environment is configured to support FIPS 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 FIPS and the module will not operate as a FIPS validated module properly.

11.1.3 Module Installation for Vendor Affirmed Platforms

Table 26 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 Server for SAP 15SP4	https://documentation.suse.com/sles/15-SP4/html/SLES-all/book-security.html
SUSE Linux Enterprise Base Container Image 15SP4	https://documentation.suse.com/smart/linux/html/concept-bci/index.html
SUSE Linux Enterprise Desktop 15SP4	https://documentation.suse.com/sled/15-SP4/html/SLED-all/book-security.html
SUSE Linux Enterprise Real Time 15SP4	https://documentation.suse.com/sle-rt/15-SP4/

Table 26 - Installation for Vendor Affirmed Platforms

Note: Per Section 7.9 in 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 Section 11.1 to configure the operational environment and install the module to be operated as a FIPS 140-3 validated module.

Table 27 lists the RPM packages that contain the FIPS validated module and the OE directory where the components are installed. The "Show version" service returns the value “Libgcrypt version 1.9.4-150400.6.8.1”, which matches the service output and the version information provided in the RPM packages where the module is distributed, and map to version 3.2 of the cryptographic module.

Processor Architecture	RPM Packages
Intel 64-bit	libgcrypt20-**1.9.4-150400.6.8.1**.x86_64.rpm libgcrypt20-hmac-**1.9.4-150400.6.8.1**.x86_64.rpm
AMD 64-bit	libgcrypt20-**1.9.4-150400.6.8.1**.x86_64.rpm libgcrypt20-hmac-**1.9.4-150400.6.8.1**.x86_64.rpm

Processor Architecture	RPM Packages
IBM z15	libgcrypt20-**1.9.4-150400.6.8.1** .s390x.rpm libgcrypt20-hmac-**1.9.4-150400.6.8.1** .s390x.rpm
ARMv8 64-bit	libgcrypt20-**1.9.4-150400.6.8.1** .aarch64.rpm libgcrypt20-hmac-**1.9.4-150400.6.8.1** .aarch64.rpm
IBM Power10 64-bit	libgcrypt20-**1.9.4-150400.6.8.1** .ppc64le.rpm libgcrypt20-hmac-**1.9.4-150400.6.8.1** .ppc64le.rpm

Table 27 – RPM packages

11.3 Non-Administrator Guidance

11.3.1 Memory Management

The user shall only use the memory management functions provided by the libgcrypt API. Critical security parameters (e.g., keys) which are used as input or output parameters shall be managed using the gcry_malloc_secure(), gcry_calloc_secure() and gcry_free() functions.

The function gcry_set_allocation_handler() shall not be used; the user shall not change the libgcrypt memory handlers. The use of this API function implies that the cryptographic module is being executed in an invalid configuration.

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 not possess persistent storage of SSPs, so further sanitization steps are not required.

12 Mitigation of Other Attacks

12.1 Attack List

The module implements blinding against RSA timing attacks.

RSA is vulnerable to timing attacks. In a setup where attackers can measure the time of RSA decryption or signature operations, blinding must be used to protect the RSA operation from that attack.

By default, the module uses the following blinding technique: instead of using the RSA decryption directly, a blinded value y = x re mod n is decrypted and the unblinded value x' = y' r–1 mod n returned. The blinding value r is a random value with the size of the modulus n.

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
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter
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
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
GMAC	Galois Counter Mode Message Authentication Code
HKDF	HMAC-based Key Derivation Function
HMAC	Keyed-Hash Message Authentication Code
IKE	Internet Key Exchange
KAS	Key Agreement Scheme
KAT	Known Answer Test
KDA	Key Derivation Algorithm
KDF	Key Derivation Function

KTS	Key Transport Scheme
KW	Key Wrap
KWP	Key Wrap with Padding
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
PBKDF2	Password-based Key Derivation Function v2
PKCS	Public Key Cryptography Standard
PRF	Pseudo-Random Function
PSP	Public Security Parameter
PSS	Probabilistic Signature Scheme
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSH	Secure Shell
SSP	Sensitive Security Parameter
TLS	Transport Layer Security
XOF	Extendable Output Function
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

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 2024 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 800-38E	**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
SP 800-38F	**Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 https://doi.org/10.6028/NIST.SP.800-38F
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

FIPS 186-4	**Digital Signature Standard (DSS)** July 2013 https://doi.org/10.6028/NIST.FIPS.186-4
SP 800-132	**Recommendation for Password-Based Key Derivation Part 1: Storage Applications** December 2010 https://doi.org/10.6028/NIST.SP.800-132
SP 800-90A 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
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1** February 2003 https://www.ietf.org/rfc/rfc3447.txt