Red Hat, Inc.

Red Hat Enterprise Linux 9 libgcrypt

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: 1.0

Last update: 2024-07-30

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 1.10.0-8b6840b590cedd43 of the Red Hat Enterprise Linux 9 libgcrypt. 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.

1.2 Security Levels

Section	Security Level
1	1
2	1
3	1
4	1
5	1
6	1
7	N/A
8	N/A
9	1
10	1
11	1
12	1
	1

Table 1: Security Levels

1.3 Additional Information

This Security Policy describes the features and design of the module named Red Hat Enterprise Linux 9 libgcrypt using the terminology contained in the FIPS 140-3 specification. The FIPS 140-3 Security Requirements for Cryptographic Module specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST/CCCS Cryptographic Module Validation Program (CMVP) validates cryptographic module to FIPS 140-3. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

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.

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.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Red Hat Enterprise Linux 9 libgcrypt (hereafter referred to as “the module”) is a software library implementing general purpose cryptographic algorithms. The module provides cryptographic services to applications running in the user space of the underlying operating system through an application program interface (API).

Module Type: Software

Module Embodiment: MultiChipStand

Module Characteristics:

Cryptographic Boundary:

The module consists of the shared library file (i.e. libgcrypt.so.20.4.0) which constitutes the cryptographic boundary. The block diagram in Figure 1 shows the cryptographic boundary of the module, its interfaces with the operational environment and the flow of information between the module and operator.

Tested Operational Environment’s Physical Perimeter (TOEPP):

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

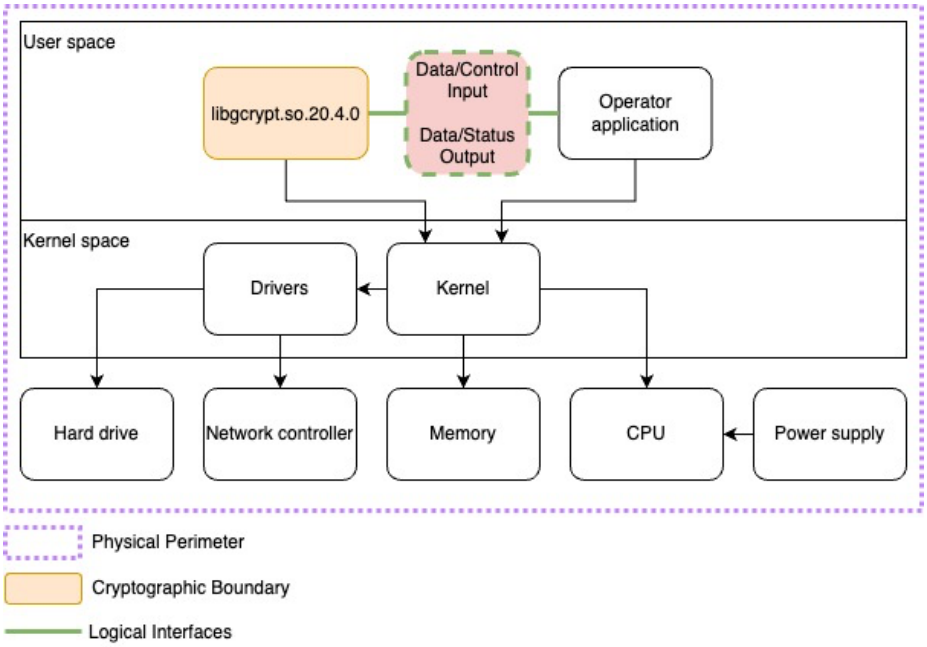

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:
N/A for this module.

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
/usr/lib64/libgcrypt.so.20.4.0	1.10.0-8b6840b590cedd43	N/A	HMAC-SHA-256

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

Tested Module Identification – Hybrid Disjoint Hardware:
N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Red Hat Enterprise Linux 9	Dell PowerEdge R440	Intel(R) Xeon(R) Silver 4216	AES-NI, SHA extensions (PAA)	N/A	1.10.0-8b6840b590cedd43
Red Hat Enterprise Linux 9	IBM z16 3931-A01	IBM z16	CPACF (PAI)	N/A	1.10.0-8b6840b590cedd43
Red Hat Enterprise Linux 9 with PowerVM FW1040.00 with VIOS 3.1.3.00	IBM 9080-HEX	IBM POWER10	ISA (PAA)	N/A	1.10.0-8b6840b590cedd43

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

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

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

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

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

2.3 Excluded Components

The module does not claim any excluded components.

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

Mode Change Instructions and Status:

When the module starts up successfully, after passing all the pre-operational self-test and the cryptographic algorithms self-tests (CASTs), the module is operating in the approved mode of operation by default and can only be transitioned into the non-approved mode by calling one of the non-approved services listed in the Non-Approved Services table. The module will transition back to approved mode when approved service is called. Section 4 provides details on the service indicator implemented by the module. The service indicator identifies when an approved service is called.

Degraded Mode Description:

The module does not implement a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3757	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A3757	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A3757	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A3757	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3757	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A3757	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A3757	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-KW	A3757	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A3757	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-XTS Testing Revision 2.0	A3757	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A3757	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-4)	A3757	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3757	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3757	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3757	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
Hash DRBG	A3757	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A3757	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA3-384	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A3757	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
PBKDF	A3757	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
RSA KeyGen (FIPS186-4)	A3757	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A3757	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-2)	A3757	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4
RSA SigVer (FIPS186-4)	A3757	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
SHA-1	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA3-224	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202

Algorithm	CAVP Cert	Properties	Reference
SHA3-384	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-512	A3757	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHAKE-128	A3757	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A3757	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
HMAC-SHA-1	A3758	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
SHA-1	A3758	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
AES-CBC	A3759	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A3759	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A3759	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A3759	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3759	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A3759	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A3759	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-KW	A3759	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A3759	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A3759	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A3759	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-4)	A3759	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3759	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3759	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
		384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	
ECDSA SigVer (FIPS186-4)	A3759	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
Hash DRBG	A3759	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A3759	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A3759	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3759	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3759	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3759	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3759	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A3759	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A3759	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
PBKDF	A3759	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
RSA KeyGen (FIPS186-4)	A3759	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A3759	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-2)	A3759	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4
RSA SigVer (FIPS186-4)	A3759	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
SHA-1	A3759	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-224	A3759	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A3759	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A3759	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A3759	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A3759	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A3759	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
AES-CBC	A3760	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A3760	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A3760	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A3760	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3760	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A3760	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A3760	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-KW	A3760	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A3760	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A3760	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A3760	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-4)	A3760	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3760	Curve - P-224, P-256, P-384, P-521	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigGen (FIPS186-4)	A3760	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3760	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
Hash DRBG	A3760	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A3760	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A3760	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3760	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3760	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3760	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3760	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A3760	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A3760	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
PBKDF	A3760	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
RSA KeyGen (FIPS186-4)	A3760	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A3760	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-2)	A3760	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4
RSA SigVer (FIPS186-4)	A3760	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
SHA-1	A3760	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A3760	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A3760	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A3760	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A3760	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A3760	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A3760	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
ECDSA KeyGen (FIPS186-4)	A3761	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3761	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3761	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3761	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
Hash DRBG	A3761	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A3761	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-256	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A3761	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
PBKDF	A3761	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
RSA KeyGen (FIPS186-4)	A3761	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A3761	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-2)	A3761	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4
RSA SigVer (FIPS186-4)	A3761	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
SHA-1	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-512	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA3-224	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-512	A3761	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHAKE-128	A3761	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A3761	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
AES-CBC	A3762	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A3762	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A3762	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A3762	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3762	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A3762	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A3762	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-KW	A3762	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A3762	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A3762	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E

Algorithm	CAVP Cert	Properties	Reference
Counter DRBG	A3762	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-4)	A3762	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3762	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3762	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3762	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
Hash DRBG	A3762	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A3762	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A3762	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
PBKDF	A3762	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
RSA KeyGen (FIPS186-4)	A3762	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A3762	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-2)	A3762	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4
RSA SigVer (FIPS186-4)	A3762	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
SHA-1	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA3-224	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202

Algorithm	CAVP Cert	Properties	Reference
SHA3-512	A3762	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHAKE-128	A3762	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A3762	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
AES-CBC	A4675	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A4675	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A4675	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A4675	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4675	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A4675	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
SHA2-256	A4675	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4675	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
AES-CBC	A4676	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A4676	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A4676	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A4676	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4676	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A4676	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
SHA2-256	A4676	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4676	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Table 6: Approved Algorithms

The above table lists all approved cryptographic algorithms of the module, including specific key lengths employed for approved services, and implemented modes or methods of operation of the algorithms.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
Cryptographic Key Generation (CKG) with RSA	RSA (FIPS 186-4):2048, 3072, 4096 with 112, 128, 149 bits of strength	libgcrypt (64 bit) (No Acceleration)	SP 800-133Rev2
Cryptographic Key Generation (CKG) with ECDSA	ECDSA:P-224, P-256, P-384, P-521 with 112, 128, 192, 256 bits of strength	libgcrypt (64 bit) (No Acceleration)	SP 800-133Rev2

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:
N/A for this module.

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

Non-Approved, Allowed Algorithms with No Security Claimed:
N/A for this module.

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

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
MD5	Message Digest
ECDH	Shared Secret Computation
AES-GCM, AES-GCM-SIV, AES-OCB, AES-EAX	Symmetric encryption and decryption
RSA	Signature generation/verification primitives
RSA	Encryption/decryption primitives
RSA using public key flags not listed in section 4.8	Key generation
RSA using public key flags not listed in section 4.8	Signature generation/verification
ECDSA	Signature generation/verification primitives
ECDSA using public key flags not listed in section 4.8	Key generation
ECDSA using public key flags not listed in section 4.8	Signature generation/verification

Table 8: Non-Approved, Not Allowed Algorithms

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

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric encryption and decryption	BC-UnAuth	Encryption, decryption using AES	Keys:128, 192, 256-bit keys with 128-256 bits key strength	AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB8 AES-CFB8 AES-CFB8 AES-CFB8 AES-CFB8 AES-CFB8 AES-CTR AES-CTR AES-CTR AES-CTR AES-CTR AES-CTR AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-OFB AES-OFB AES-OFB AES-OFB
Message authentication generation using AES CMAC	MAC	Message authentication generation using AES CMAC	AES-CMAC keys:128, 192, 256-bit keys with 128, 192, 256 bits of strength	AES-CMAC AES-CMAC AES-CMAC AES-CMAC
Key wrapping with AES	KTS-Wrap	Key wrapping with AES	Keys:128, 192, 256-bit keys with 128-256 bits of strength Compliance:Compliant with IG D.G AES Mode:KW, CCM	AES-KW AES-KW AES-KW AES-KW AES-CCM AES-CCM AES-CCM AES-CCM

Name	Type	Description	Properties	Algorithms
Key unwrapping with AES	KTS-Wrap	Key unwrapping with AES	Keys:128, 192, 256-bit keys with 128-256 bits of strength Compliance:Compliant with IG D.G AES Mode:KW, CCM	AES-KW AES-KW AES-KW AES-KW AES-CCM AES-CCM AES-CCM AES-CCM
Authenticated symmetric encryption and decryption	BC-Auth	Encryption, decryption using AES CCM	Keys:128, 192, 256 bits with 128, 192, 256 bits of strength	AES-CCM AES-CCM AES-CCM AES-CCM
Symmetric encryption and decryption with AES XTS (for data storage)	BC-UnAuth	Encryption, decryption using AES XTS (for data storage)	Keys:128, 256-bit keys with 128, 256 bits of strength	AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0
Random number generation with CTR_DRBG	DRBG	Random number generation using CTR_DRBG	CTR_DRBG:AES-128, AES-192, AES-256 with DF, with/without PR	Counter DRBG Counter DRBG Counter DRBG Counter DRBG
Random number generation with HMAC_DRBG	DRBG	Random number generation using HMAC_DRBG	HMAC_DRBG:SHA-1, SHA-256, SHA-512 with/without PR	HMAC DRBG HMAC DRBG HMAC DRBG HMAC DRBG HMAC DRBG
Random number generation with Hash_DRBG	DRBG	Random number generation using Hash_DRBG	Hash_DRBG:SHA-1, SHA-256, SHA-512 with/without PR	Hash DRBG Hash DRBG Hash DRBG Hash DRBG Hash DRBG
Key pair generation with ECDSA	CKG	Key pair generation using ECDSA	Curves:P-224, P-256, P-384, P-521 with 112, 128, 192, 256 bits of strength	ECDSA KeyGen (FIPS186-4) ECDSA

Name	Type	Description	Properties	Algorithms
		IG D.H compliant	Method:B.4.2 Testing Candidates	KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4)
Key pair generation with RSA	CKG	Key pair generation using RSA IG D.H compliant	Keys:2048, 3072, 4096 with 112, 128, 149 bits of strength Method:B.3.3 Random Probable Primes	RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4)
Public key verification with ECDSA	AsymKeyPair-KeyVer	Public key verification using ECDSA	Curves:P-224, P-256, P-384, P-521 with 112, 128, 192, 256 bits of strength	ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4)
Digital signature generation with ECDSA	DigSig-SigGen	Digital signature generation using ECDSA	Curves:P-224, P-256, P-384, P-521 with 112, 128, 192, 256 bits of strength Hash:SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4)
Digital signature generation with RSA	DigSig-SigGen	Digital signature generation using RSA	Padding:PKCS#1v1.5, PSS Hash:SHA-224, SHA-256, SHA-384, SHA-512, SHA2-512/224, SHA2-512/256 Keys:2048, 3072,	RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4)

Name	Type	Description	Properties	Algorithms
			4096 with 112, 128, 149 bits of strength	RSA SigGen (FIPS186-4)
Digital signature verification with ECDSA	DigSig-SigVer	Digital signature verification using ECDSA	Curves:P-224, P-256, P-384, P-521 with 112, 128, 192, 256 bits of strength Hash:SHA2-224, SHA2- 256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4)
FIPS 186-4 Digital signature verification with RSA	DigSig-SigVer	FIPS 186-4 Digital signature verification using RSA	Padding:PKCS#1v1.5, PSS Hash:SHA-224, SHA-256, SHA-384, SHA-512, SHA2-512/224, SHA2-512/256 Keys:2048, 3072, 4096 with 112, 128, 149 bits of strength	RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4)
FIPS 186-2 Digital signature verification with RSA	DigSig-SigVer	FIPS 186-2 Digital signature verification using RSA Only for Legacy use	Padding:PKCS#1v1.5, PSS Hash:SHA-224, SHA-256, SHA-384, SHA-512 Keys:1024, 1536 with 80, 92 bits of strength Compliance:FIPS 186-2, IG C.F	RSA SigVer (FIPS186-2) RSA SigVer (FIPS186-2) RSA SigVer (FIPS186-2) RSA SigVer (FIPS186-2) RSA SigVer (FIPS186-2)
Message authentication code with HMAC	MAC	Message authentication code using HMAC	Keys:112, 192, 256 bits with 112-256 bits of strength Hash:SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-

Name	Type	Description	Properties	Algorithms
				256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA3-224 HMAC-SHA3-

Name	Type	Description	Properties	Algorithms
				224 HMAC-SHA3-224 HMAC-SHA3-256 HMAC-SHA3-256 HMAC-SHA3-256 HMAC-SHA3-384 HMAC-SHA3-384 HMAC-SHA3-384 HMAC-SHA3-512 HMAC-SHA3-512 HMAC-SHA3-512
Key derivation with PBKDF	PBKDF	Key derivation using PBKDF	Derived key::112 to 256 bits HMAC:SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3- 384, SHA3-512	PBKDF PBKDF PBKDF PBKDF PBKDF
Message digest with SHA-1	SHA	Message digest using SHA-1		SHA-1 SHA-1 SHA-1 SHA-1 SHA-1 SHA-1
Message digest with SHA-2	SHA	Message digest using SHA-2		SHA2-224 SHA2-224 SHA2-224 SHA2-224 SHA2-224 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-384 SHA2-384 SHA2-384

Name	Type	Description	Properties	Algorithms
				SHA2-384 SHA2-384 SHA2-512 SHA2-512 SHA2-512 SHA2-512 SHA2-512 SHA2-512 SHA2-512/224 SHA2-512/224 SHA2-512/224 SHA2-512/224 SHA2-512/224 SHA2-512/256 SHA2-512/256 SHA2-512/256 SHA2-512/256 SHA2-512/256
Message digest with SHA-3	SHA	Message digest with SHA-3		SHA3-224 SHA3-224 SHA3-224 SHA3-256 SHA3-256 SHA3-256 SHA3-384 SHA3-384 SHA3-384 SHA3-512 SHA3-512 SHA3-512 SHAKE-128 SHAKE-128 SHAKE-128 SHAKE-256 SHAKE-256 SHAKE-256

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

AES XTS

The AES algorithm in XTS mode can be only used for the cryptographic protection of data on storage devices, as specified in [SP800-38E]. 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.

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

Key derivation using SP800-132 PBKDF

The module provides password-based key derivation (PBKDF), compliant with SP800-132. The module supports option 1a from Section 5.4 of [SP800-132], in which the Master Key (MK) or a segment of it is used directly as the Data Protection Key (DPK).

In accordance with [SP800-132] and FIPS 140-3 IG D.N, the following requirements shall be met.

- Derived keys shall only be used in storage applications. The Master Key (MK) shall not be used for other purposes. The module accepts length of the MK or DPK of 112 bits or more.
- A portion of the salt, with a length of at least 128 bits, shall be generated randomly using the SP800-90A DRBG.
- 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.
- Passwords or passphrases, used as an input for the PBKDF, shall not be used as cryptographic keys.
- The minimum length of the password or passphrase accepted by the module is 8 characters. The probability of guessing the value, assuming a worst-case scenario of all digits, is estimated to be at most 10^{-8}. Combined with the minimum iteration count as described below, this provides an acceptable trade-off between user experience and security against brute-force attacks.

The calling application shall also observe the rest of the requirements and recommendations specified in [SP800-132].

2.8 RBG and Entropy

Cert Number	Vendor Name
E47	Red Hat, Inc.

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
RHEL Userspace CPU Time Jitter RNG Entropy Source	Non-Physical	Red Hat Enterprise Linux 9 running on Dell PowerEdge R440 with Intel(R) Xeon(R) Silver 4216; Red Hat Enterprise Linux 9 running on IBM z16 3931-A01 with IBM z16; Red Hat Enterprise Linux 9 with PowerVM FW1040.00 with VIOS 3.1.3.00 running on	256-bits	225-bits	HMAC-SHA2-512 DRBG

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
		IBM 9080-HEX with IBM POWER10			

Table 11: Entropy Sources

The Module provides an SP800-90A-compliant Deterministic Random Bit Generator (DRBG) for creation of key components of asymmetric keys, and random number generation.

The seeding (and automatic reseeding) of the DRBG is done with getrandom().

The module supports the Hash_DRBG, HMAC_DRBG and CTR_DRBG. The DRBG is initialized during module initialization; the module loads by default the DRBG using the HMAC_DRBG mechanism with SHA-256 and without prediction resistance. A different DRBG mechanism can be chosen by invoking the gcry_control(GCRYCTL_DRBG_REINIT) function.

The module uses an [SP800-90B]-compliant entropy source specified in the above table. This entropy source is located within the module's physical perimeter but outside of the module's cryptographic boundary. The module obtains 384 bits to seed the DRBG and 256 bits to reseed it, respectivly corresponding to 337 bits of entropy for seeding and 225 bits for reseeding, which is not full entropy. Therefore, the module generates SSPs (e.g., keys) whose strengths are modified by available entropy.

The module performs the DRBG health tests as defined in Section 11.3 of [SP800-90A].

2.9 Key Generation

KeyGenerationSubTable From Web Cryptik KeyGenerationSubTable

The module provides an [SP800-90Arev1]-compliant Deterministic Random Bit Generator (DRBG) for the creation of key components of asymmetric keys, and random number generation.

The Cryptographic Key Generation (CKG) methods implemented in the module for Approved services in approved mode are compliant with section 5.1 of [SP800-133rev2] and with IG D.H. For generating RSA and ECDSA keys the module implements asymmetric key

generation services compliant with [FIPS186-4]. A seed (i.e., the random value) used in asymmetric key generation is directly obtained from the [SP800-90Arev1] DRBG.

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

2.10 Key Establishment

KeyAgreementSubTable From Web Cryptik KeyAgreementSubTable

KeyTransportSubTable From Web Cryptik KeyTransportSubTable

The module provides the following key transport mechanisms:

- AES key wrapping using AES-KW.
- AES key wrapping using AES-CCM.

According to Table 2: Comparable strengths in [SP 800-57rev5], the key sizes of AES provide the following security strength in approved mode of operation:

- AES key wrapping in KW mode provides 128, 192, 256-bit keys with key strength between 128-256 bits in compliance with SP800-38F and IG D.G.
- AES key wrapping using AES-CCM provides 128, 192, 256-bit keys with key strength between 128-256 bits in compliance with SP800-38F and IG D.G.

Additionally, the module supports password-based key derivation (PBKDF2). The implementation is compliant with option 1a of [SP-800-132]. Keys derived from passwords or passphrases using this method can only be used in storage applications

2.11 Industry Protocols

The module does not implement industry protocols, therefore this section is not applicable.

2.12 Additional Information

Not Applicable.

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.

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

The module does not implement a control output interface.

3.2 Trusted Channel Specification

Not Applicable.

3.3 Control Interface Not Inhibited

Not Applicable.

3.4 Additional Information

Not Applicable.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module does not support authentication.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 13: Roles

The module supports the Crypto Officer role only. This sole role is implicitly and always assumed by the operator of the module.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric encryption and decryption	Encrypt a plaintext / Decrypt a ciphertext	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHE R, ...) returns GPG_ERR_NO_ERROR	AES key, plaintext/ciphertext	Ciphertext/ plaintext	Symmetric encryption and decryption Authenticated symmetric encryption and decryption Symmetric encryption and decryption with AES XTS (for data storage)	Crypto Officer - AES keys: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Key Pair Generation with RSA	Generate a key pair	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) returns GPG_ERR_NO_ERROR	Modulus bits	RSA public key, RSA private key	Key pair generation with RSA	Crypto Officer - RSA Private Key: G,R - RSA Public Key: G,R - Intermediate key generation value: G,E,Z
Key Pair Generation with ECDSA	Generate a key pair	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) returns GPG_ERR_NO_ERROR	Curve	ECDSA public key, ECDSA private key	Key pair generation with ECDSA	Crypto Officer - ECDSA Private Key: G,R - ECDSA Public Key: G,R - Intermediate key generation value: G,E,Z
Digital signature generation with RSA	Generate a signature	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) and gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) return	RSA private key, message	Signature	Digital signature generation with RSA	Crypto Officer - RSA Private Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		GPG_ERR_NO_ERROR				
Digital signature generation with ECDSA	Generate a signature	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) and gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) return GPG_ERR_NO_ERROR	ECDSA private key, message	Signature	Digital signature generation with ECDSA	Crypto Officer - ECDSA Private Key: W,E
Digital signature verification with RSA	Verify a signature	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) and gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) return GPG_ERR_NO_ERROR	RSA public key, message, signature	Pass/fail	FIPS 186-4 Digital signature verification with RSA FIPS 186-2 Digital signature verification with RSA	Crypto Officer - RSA Public Key: W,E
Digital signature verification with ECDSA	Verify a signature	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) and gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) return GPG_ERR_NO_ERROR	ECDSA public key, message, signature	Pass/fail	Digital signature verification with ECDSA	Crypto Officer - ECDSA Public Key: W,E
Public key verification	Verify ECDSA public key	gcry_mpi_ec_curve_point() returns GPG_ERR_NO_ERROR	ECDSA public key	Pass/fail	Public key verification with ECDSA	Crypto Officer - ECDSA Public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key: W,E
Random Number Generation with CTR_DRBG/HMAC_DRBG	Generate random bitstrings from CTR_DRBG/HMAC_DRBG	gcry_randomize(), gcry_random_bytes(), gcry_random_bytes_secure() return GPG_ERR_NO_ERROR	Output length	Random bytes	Random number generation with CTR_DRBG Random number generation with HMAC_DRBG	Crypto Officer - Entropy Input: W,E - DRBG seed: G,E - DRBG internal state (V value, Key): G,W,E
Random Number Generation with Hash_DRBG	Generate random bitstrings from Hash_DRBG	gcry_randomize(), gcry_random_bytes(), gcry_random_bytes_secure() return GPG_ERR_NO_ERROR	Output length	Random bytes	Random number generation with Hash_DRBG	Crypto Officer - Entropy Input: W,E - DRBG seed: G,E - DRBG internal state (V value, C value): G,W,E
Message digest	Compute SHA hashes	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) returns GPG_ERR_NO_ERROR	Message	Digest value	Message digest with SHA-3 Message digest with SHA-1 Message digest	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					with SHA-2	
Message authentication code (MAC) with HMAC	Compute HMAC	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MAC, .. .) returns GPG_ERR_NO_ERROR	HMAC key	MAC tag	Message authentication code with HMAC	Crypto Officer - HMAC keys: W,E
Message authentication code (MAC) with CMAC	Compute AES-based CMAC	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MAC, .. .) returns GPG_ERR_NO_ERROR	AES key	MAC tag	Message authentication generation with AES-CMAC	Crypto Officer - AES keys: W,E
Key wrapping	Perform AES-based key wrapping	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER, ...) returns GPG_ERR_NO_ERROR	AES key, any CSP	Wrapped CSP	Key wrapping with AES	Crypto Officer - AES keys: W,E
Key unwrapping	Perform AES-based key unwrapping	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER, ...) returns GPG_ERR_NO_ERROR	AES key, wrapped CSP	Unwrapped CSP	Key unwrapping with AES	Crypto Officer - AES keys: W,E
Key derivation	Perform key derivation	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_KDF, ...) returns GPG_ERR_NO_ERROR	Password, salt, iteration count	Derived key	Key derivation with PBKDF	Crypto Officer - Password or passphrase: W,E - Derived key: G,R
On-demand Integrity test	Perform on-demand integrity test	N/A	N/A	Pass/fail	Message authentication	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					code with HMAC	
Show status	Show module status	N/A	N/A	Module status	None	Crypto Officer
Zeroization	Zeroize all SSPs	N/A	Any SSP	N/A	None	Crypto Officer
Self-tests	Perform self-tests	N/A	N/A	Pass/fail	None	Crypto Officer
Show module name and version	Show module name and version	N/A	N/A	Module name and version information	None	Crypto Officer

Table 14: Approved Services

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

- **G** = **Generate**: The module generates or derives the SSP.
- **R** = **Read**: The SSP is read from the module (e.g., the SSP is output).
- **W** = **Write**: The SSP is updated, imported, or written to the module.
- **E** = **Execute**: The module uses the SSP in performing a cryptographic operation.
- **Z** = **Zeroise**: The module zeroises the SSP.
- **N/A**: the calling application does not access any CSP or key during its operation.

The details of the approved cryptographic algorithms including the CAVP certificate numbers can be found in the Approved Algorithm table. In order to check whether it utilizes an approved security function or not, the operator is responsible to invoke the gcry_control() API along with dedicated controls in the form of API input parameters.

The module implements the following controls depending on the requested service:

1. GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER - For symmetric algorithms and the related modes.
2. GCRYCTL_FIPS_SERVICE_INDICATOR_KDF - For KDF operations.

3. GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS - For asymmetric operations.[1]
4. GCRYCTL_FIPS_SERVICE_INDICATOR_MD - For digest operations.
5. GCRYCTL_FIPS_SERVICE_INDICATOR_MAC - For MAC operations.

In addition to that, for the below-mentioned services, the approved service indicator corresponds to the GPG_ERR_NO_ERROR returned from listed functions in the indicator column below. They don't use gcry_control() API:

1. *Random number generation* service: gcry_randomize(), gcry_random_bytes(), gcry_random_bytes_secure().
2. *Public key validation* service: gcry_mpi_ec_curve_point().

For all approved services, GPG_ERR_NO_ERROR (i.e., "0") return code indicates the service is approved. In case the above-mentioned controls are used in conjunction, the operator is responsible to check that all of the called functions return GPG_ERR_NO_ERROR (i.e., "0"). For all non-approved services, "non-zero" return code indicates the service is not approved.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Symmetric encryption/decryption	AES encryption/decryption using non-approved AES modes	AES-GCM, AES-GCM-SIV, AES-OCB, AES-EAX	CO
Message digest	Non-approved message digest	MD5	CO
Shared Secret Computation	ECDH Shared Secret Computation	ECDH	CO
Key generation with RSA	Generate RSA key pairs using public key flags not listed in section 4.8.	RSA using public key flags not listed in section 4.8 RSA using public key flags not listed in section 4.8	CO
Key generation with ECDSA	Generate ECDSA key pairs using public key flags not listed in section 4.8.	ECDSA using public key flags not listed in section 4.8 ECDSA using public key flags not listed in section 4.8	CO
Digital signature generation/verification with RSA	Generate/verify a signature using RSA Signature generation/verification primitives	RSA	CO
Digital signature generation/verification with ECDSA	Generate/verify a signature using ECDSA Signature generation/verification primitives	ECDSA	CO

[1] The list of public key flags allowed in approved mode of operation is described in Section 4.8 Additional Information.

Name	Description	Algorithms	Role
Asymmetric encryption/decryption	Perform encryption/decryption using RSA encryption/decryption primitives	RSA	CO

Table 15: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not have the capability of loading software or firmware from an external source.

4.6 Bypass Actions and Status

Not Applicable.

4.7 Cryptographic Output Actions and Status

Not Applicable.

4.8 Additional Information

Below are listed the approved public key flags for an input s-expression:

curve	d	data	e	ecdsa	flags	sig-val
genkey	hash	n	nbits	pkcs1	private-key	value
pss	public-key	q	r	raw	rsa	salt-length
rsa-use-e	s					

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified comparing the HMAC-SHA-256 value calculated at run time with the HMAC-SHA-256 value embedded in the module's ELF header 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.

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 relying on the gcry_control(GCRYCTL_SELFTEST) API function call or by powering-off and reloading the module. 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 gcry_control(GCRYCTL_OPERATIONAL_P). The function will return TRUE if the module is in the operational state, FALSE if the module is in the Error state.

5.3 Open-Source Parameters

Not Applicable.

5.4 Additional Information

Not Applicable.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

The module should be compiled and installed as stated in section 11. The user should confirm that the module is installed correctly by running:

1. fips-mode-setup --check command to verify that the system is operating in Approved mode
2. check the output of the gcry_get_config() API, which should output *Red Hat Enterprise Linux 9 libgcrypt 1.10.0-8b6840b590cedd43*

The module does not support concurrent operators.

6.2 Configuration Settings and Restrictions

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

6.3 Additional Information

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

7 Physical Security

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

7.1 Mechanisms and Actions Required

Not Applicable.
N/A for this module.

7.2 User Placed Tamper Seals

Not Applicable.

Number:

Placement:

Surface Preparation:

Operator Responsible for Securing Unused Seals:

Part Numbers:

7.3 Filler Panels

Not Applicable.

7.4 Fault Induction Mitigation

Not Applicable.

7.5 EFP/EFT Information

Not Applicable.

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature			
HighTemperature			
LowVoltage			
HighVoltage			

Table 16: EFP/EFT Information

7.6 Hardness Testing Temperature Ranges

Not Applicable.

Temperature Type	Temperature
LowTemperature	
HighTemperature	

Table 17: Hardness Testing Temperatures

7.7 Additional Information

Not Applicable.

8 Non-Invasive Security

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

8.1 Mitigation Techniques

Not Applicable.

8.2 Effectiveness

Not Applicable.

8.3 Additional Information

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. The module does not perform persistent storage of SSPs	Dynamic

Table 18: Storage Areas

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

9.2 SSP Input-Output Methods

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

Table 19: 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 [FIPS140-3_IG] 9.5.A, according to the “CM Software to/from App via TOEPP Path” entry on the Key Establishment Table.

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. The completion of the zeroization routine	By calling the cipher related zeroization API which are the following: gcry_free(), gcry_cipher_close(), gcry_mac_close(), gcry_sexp_release(), gcry_mpi_release(), gcry_ctx_release(), gcry_mpi_point_release(),

Zeroization Method	Description	Rationale	Operator Initiation
		indicates that the zeroization procedure succeeded.	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 20: SSP Zeroization Methods

The memory occupied by SSPs is allocated by regular memory allocation operating system calls. The application that is acting as the CO is responsible for calling the appropriate zeroization functions provided in the module's API and listed in the above table. Calling gcry_free(), which will zeroize the SSPs and also invoke the corresponding API functions listed in the above table to zeroize SSPs. The zeroization functions overwrite the memory occupied by SSPs with “zeros” and deallocate the memory with the regular memory deallocation operating system call. In case of abnormal termination, or swap in/out of a physical memory page of a process, the keys in physical memory are overwritten by the Linux kernel before the physical memory is allocated to another process. The completion of a zeroization routine(s) will indicate that a zeroization procedure succeeded.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES keys	AES key used for encryption, decryption, and computing MAC tags	AES-XTS: 128, 256; Other modes: 128, 192, 256 - AES-XTS: 128, 256; Other modes: 128, 192, 256	Symmetric key - CSP			Symmetric encryption and decryption Message authentication generation with AES-CMAC Authenticated symmetric encryption and decryption Symmetric encryption

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						and decryption with AES XTS (for data storage)
HMAC keys	HMAC key used for computing MAC tags	112-256 bits - 112-256 bits	Symmetric key - CSP			Message authentication code with HMAC
RSA Private Key	Private key used for RSA signature generation	2048, 3072, 4096 bits - 112, 128, 149 bits	Private key - CSP	Key pair generation with RSA		Digital signature generation with RSA
RSA Public Key	Public key used for RSA signature verification	2048, 3072, 4096 bits bits - 112, 128, 149 bits	Public key - PSP	Key pair generation with RSA		FIPS 186-4 Digital signature verification with RSA FIPS 186-2 Digital signature verification with RSA
ECDSA Private Key	Private key used for ECDSA signature generation	P-224, P-256, P-384, P-521 - 112, 128, 192, 256 bits	Private key - CSP	Key pair generation with ECDSA		Public key verification with ECDSA Digital signature generation with ECDSA
ECDSA Public Key	Public key used for ECDSA signature verification	P-224, P-256, P-384, P-521 - 112, 128, 192, 256 bits	Public key - PSP	Key pair generation with ECDSA		Digital signature verification with ECDSA
Intermediate key generation value	Intermediate key pair generation value generated during key generatio	112-256 - 112-256 bits	Intermediate value - CSP	Key pair generation with RSA Key pair generation with ECDSA		Key pair generation with RSA Key pair generation with ECDSA

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	n and key derivation services (SP 800-133r2 Section 4, 5.1, and 5.2)					
Password or passphrase	Password used to derive symmetric keys	Minimum of 8 character - N/A	Password - CSP			Key derivation with PBKDF
Derived key	Symmetric key derived from a key derivation key, shared secret, or password	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation with PBKDF		
Entropy Input	Entropy input used to seed the DRBG (IG D.L compliant)	128-384 bits - 112-337 bits	Entropy input - CSP			Random number generation with CTR_DRBG Random number generation with HMAC_DRBG Random number generation with Hash_DRBG
DRBG internal state (V value, C value)	Internal state of the Hash_DRBG (IG D.L compliant)	880, 1776 bits - 128, 256 bits	Internal state - CSP	Random number generation with Hash_DRBG		Random number generation with Hash_DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG internal state (V value, Key)	Internal state of the CTR_DRBG and HMAC_DRBG (IG D.L compliant)	CTR_DRBG: 256, 320, 384 bits; HMAC_DRBG: 320, 512, 1024 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256 bits	Internal state - CSP	Random number generation with CTR_DRBG Random number generation with HMAC_DRBG		Random number generation with CTR_DRBG Random number generation with HMAC_DRBG
DRBG seed	DRBG seed derived from entropy input as defined in SP 800-90Ar1 (IG D.L compliant)	CTR_DRBG: 256, 320, 384 bits; HMAC_DBRG: 440, 880 bits;Hash_DRBG: 440, 880 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256 bits; Hash_DRBG: 128, 256 bits;	Seed - CSP	Random number generation with CTR_DRBG Random number generation with HMAC_DRBG Random number generation with Hash_DRBG		Random number generation with CTR_DRBG Random number generation with HMAC_DRBG Random number generation with Hash_DRBG

Table 21: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES keys	API input parameters (plaintext)	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	
HMAC keys	API input parameters (plaintext)	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
RSA Private Key	API input parameters (plaintext) API output parameters (plaintext)	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	RSA Public Key:Paired With DRBG internal state (V value, Key):Generated from Intermediate key generation value:Generated from
RSA Public Key	API input parameters (plaintext) API output parameters (plaintext)	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	RSA Private Key:Paired With DRBG internal state (V value, C value):Generated from Intermediate key generation value:Generated from
ECDSA Private Key	API input parameters (plaintext) API output parameters (plaintext)	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	ECDSA Public Key:Paired With DRBG internal state (V value, C value):Generated from Intermediate key generation value:Generated from
ECDSA Public Key	API input parameters (plaintext) API output parameters (plaintext)	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	ECDSA Private Key:Paired With DRBG internal state (V value, C value):Generated from Intermediate key generation value:Generated from
Intermediate key generation value		RAM:Plaintext	From service invocation until cipherhandle is freed	Automatic	RSA Private Key:Generates RSA Public Key:Generates ECDSA Private Key:Generates ECDSA Public Key:Generates

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Password or passphrase	API input parameters (plaintext)	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	Derived key:Derivation of
Derived key	API output parameters (plaintext)	RAM:Plaintext	From service invocation until cipherhandle is freed	Wipe and Free memory block allocated Module Reset	Password or passphrase:Derived From
Entropy Input		RAM:Plaintext	From service invocation until cipherhandle is freed	Automatic	DRBG seed:Derivation of
DRBG internal state (V value, C value)		RAM:Plaintext	From service invocation until cipherhandle is freed	Automatic	DRBG seed:Generated from
DRBG internal state (V value, Key)		RAM:Plaintext	From service invocation until cipherhandle is freed	Automatic	DRBG seed:Generated from
DRBG seed		RAM:Plaintext	From service invocation until cipherhandle is freed	Automatic	Entropy Input:Derived From DRBG internal state (V value, C value):Generation of DRBG internal state (V value, Key):Generation of

Table 22: SSP Table 2

The tables above summarizes the Sensitive Security Parameters (SSPs) that are used by the cryptographic services implemented in the module.

9.5 Transitions

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes, starting January 1, 2030.

9.6 Additional Information

Not Applicable.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A3760)	256-bit key	Message authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for /usr/lib64/libgcrypt.so.20.4.0

Table 23: Pre-Operational Self-Tests

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

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A3757)	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3759)	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3760)	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3762)	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A4675)	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4676)	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3757)	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3759)	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3760)	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3762)	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4675)	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4676)	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CMAC (A3757)	128-bit key MAC generation, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-CMAC (A3759)	128-bit key MAC	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	generation, encrypt					integrity test
AES-CMAC (A3760)	128-bit key MAC generation, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-CMAC (A3762)	128-bit key MAC generation, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
Counter DRBG (A3757)	AES 128-bit key with DF, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Counter DRBG (A3759)	AES 128-bit key with DF, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Counter DRBG (A3760)	AES 128-bit key with DF, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Counter DRBG (A3762)	AES 128-bit key with DF, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A3757)	SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A3759)	SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A3760)	SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Hash DRBG (A3761)	SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A3762)	SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A3757)	SHA-1 without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A3759)	SHA-1 without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A3760)	SHA-1 without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A3761)	SHA-1 without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A3762)	SHA-1 without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
HMAC DRBG (A3757)	HMAC-SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
HMAC DRBG (A3759)	HMAC-SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
HMAC DRBG (A3760)	HMAC-SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
HMAC DRBG (A3761)	HMAC-SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
HMAC DRBG (A3762)	HMAC-SHA2-256 with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3757)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3759)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3760)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3761)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A3762)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3757)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3759)	P-256 and SHA2-256	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
ECDSA SigVer (FIPS186-4) (A3760)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3761)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3762)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3757)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3759)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3760)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3761)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3762)	P-256 and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3757)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3758)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
HMAC-SHA-1 (A3759)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3760)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3761)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3762)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3757)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3759)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3760)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3761)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3762)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-256 (A3757)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3759)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3760)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3761)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3762)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3757)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3759)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3760)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3761)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3762)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
HMAC-SHA2-512 (A3757)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3759)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3760)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3761)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3762)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-224 (A3757)	SHA3-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-224 (A3761)	SHA3-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-224 (A3762)	SHA3-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-256 (A3757)	SHA3-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA3-256 (A3761)	SHA3-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-256 (A3762)	SHA3-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-384 (A3757)	SHA3-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-384 (A3761)	SHA3-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-384 (A3762)	SHA3-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-512 (A3757)	SHA3-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-512 (A3761)	SHA3-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-512 (A3762)	SHA3-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A3757)	PKCS#1 v1.5 with 2048-bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen	PKCS#1 v1.5 with 2048-bit	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
(FIPS186-4) (A3759)	key and SHA2-256					integrity test
RSA SigGen (FIPS186-4) (A3760)	PKCS#1 v1.5 with 2048-bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A3761)	PKCS#1 v1.5 with 2048-bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A3762)	PKCS#1 v1.5 with 2048-bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A3757)	PKCS#1 v1.5 with 2048-bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A3759)	PKCS#1 v1.5 with 2048-bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A3760)	PKCS#1 v1.5 with 2048-bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A3761)	PKCS#1 v1.5 with 2048-bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A3762)	PKCS#1 v1.5 with 2048-bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
SHA-1 (A3757)	Message digest	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 (A3758)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3759)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3760)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3761)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3762)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3757)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3759)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3760)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3761)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3762)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
SHA2-256 (A3757)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3759)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3760)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3761)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3762)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A4675)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A4676)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3757)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3759)	Message digest	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
SHA2-384 (A3760)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3761)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3762)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3757)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3759)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3760)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3761)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3762)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4675)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4676)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
PBKDF (A3757)	SHA-1 password length 24 characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3759)	SHA-1 password length 24 characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3760)	SHA-1 password length 24 characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3761)	SHA-1 password length 24 characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3762)	SHA-1 password length 24	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits					integrity test
PBKDF (A3757)	SHA2-256 password length 24 characters, master key length of 320 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3759)	SHA2-256 password length 24 characters, master key length of 320 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3760)	SHA2-256 password length 24 characters, master key length of 320 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A3761)	SHA2-256 password length 24 characters, master key length of 320 bits, iteration count of	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	4096, and salt length of 288 bits					
PBKDF (A3762)	SHA2-256 password length 24 characters, master key length of 320 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
RSA KeyGen (FIPS186-4) (A3757)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A3759)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A3760)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A3761)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A3762)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A3757)	SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A3759)	SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A3760)	SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation
ECDSA KeyGen	SHA2-256	PCT	PCT	Key pair generation	Signature generation	Key pair generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
(FIPS186-4) (A3761)				is successful	and verification	
ECDSA KeyGen (FIPS186-4) (A3762)	SHA2-256	PCT	PCT	Key pair generation is successful	Signature generation and verification	Key pair generation

Table 24: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A3760)	Message authentication	SW/FW Integrity	On demand	Manually

Table 25: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A3757)	KAT	CAST	On Demand	Manually
AES-ECB (A3759)	KAT	CAST	On Demand	Manually
AES-ECB (A3760)	KAT	CAST	On Demand	Manually
AES-ECB (A3762)	KAT	CAST	On Demand	Manually
AES-ECB (A4675)	KAT	CAST	On Demand	Manually
AES-ECB (A4676)	KAT	CAST	On Demand	Manually
AES-ECB (A3757)	KAT	CAST	On Demand	Manually
AES-ECB (A3759)	KAT	CAST	On Demand	Manually
AES-ECB (A3760)	KAT	CAST	On Demand	Manually
AES-ECB (A3762)	KAT	CAST	On Demand	Manually
AES-ECB (A4675)	KAT	CAST	On Demand	Manually
AES-ECB (A4676)	KAT	CAST	On Demand	Manually
AES-CMAC (A3757)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CMAC (A3759)	KAT	CAST	On Demand	Manually
AES-CMAC (A3760)	KAT	CAST	On Demand	Manually
AES-CMAC (A3762)	KAT	CAST	On Demand	Manually
Counter DRBG (A3757)	KAT	CAST	On Demand	Manually
Counter DRBG (A3759)	KAT	CAST	On Demand	Manually
Counter DRBG (A3760)	KAT	CAST	On Demand	Manually
Counter DRBG (A3762)	KAT	CAST	On Demand	Manually
Hash DRBG (A3757)	KAT	CAST	On Demand	Manually
Hash DRBG (A3759)	KAT	CAST	On Demand	Manually
Hash DRBG (A3760)	KAT	CAST	On Demand	Manually
Hash DRBG (A3761)	KAT	CAST	On Demand	Manually
Hash DRBG (A3762)	KAT	CAST	On Demand	Manually
Hash DRBG (A3757)	KAT	CAST	On Demand	Manually
Hash DRBG (A3759)	KAT	CAST	On Demand	Manually
Hash DRBG (A3760)	KAT	CAST	On Demand	Manually
Hash DRBG (A3761)	KAT	CAST	On Demand	Manually
Hash DRBG (A3762)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3757)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3759)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3760)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3761)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3762)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigGen (FIPS186-4) (A3757)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3759)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3760)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3761)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3762)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3757)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3759)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3760)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3761)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3762)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3757)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3759)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3760)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3761)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3762)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA-1 (A3757)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3758)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3759)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3760)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3761)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3762)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3757)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3759)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3760)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3761)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3762)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3757)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3759)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3760)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3761)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3762)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3757)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3759)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3760)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3761)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3762)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3757)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-512 (A3759)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3760)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3761)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3762)	KAT	CAST	On Demand	Manually
HMAC-SHA3-224 (A3757)	KAT	CAST	On Demand	Manually
HMAC-SHA3-224 (A3761)	KAT	CAST	On Demand	Manually
HMAC-SHA3-224 (A3762)	KAT	CAST	On Demand	Manually
HMAC-SHA3-256 (A3757)	KAT	CAST	On Demand	Manually
HMAC-SHA3-256 (A3761)	KAT	CAST	On Demand	Manually
HMAC-SHA3-256 (A3762)	KAT	CAST	On Demand	Manually
HMAC-SHA3-384 (A3757)	KAT	CAST	On Demand	Manually
HMAC-SHA3-384 (A3761)	KAT	CAST	On Demand	Manually
HMAC-SHA3-384 (A3762)	KAT	CAST	On Demand	Manually
HMAC-SHA3-512 (A3757)	KAT	CAST	On Demand	Manually
HMAC-SHA3-512 (A3761)	KAT	CAST	On Demand	Manually
HMAC-SHA3-512 (A3762)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3757)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3759)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3760)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3761)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigGen (FIPS186-4) (A3762)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3757)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3759)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3760)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3761)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A3762)	KAT	CAST	On Demand	Manually
SHA-1 (A3757)	KAT	CAST	On Demand	Manually
SHA-1 (A3758)	KAT	CAST	On Demand	Manually
SHA-1 (A3759)	KAT	CAST	On Demand	Manually
SHA-1 (A3760)	KAT	CAST	On Demand	Manually
SHA-1 (A3761)	KAT	CAST	On Demand	Manually
SHA-1 (A3762)	KAT	CAST	On Demand	Manually
SHA2-224 (A3757)	KAT	CAST	On Demand	Manually
SHA2-224 (A3759)	KAT	CAST	On Demand	Manually
SHA2-224 (A3760)	KAT	CAST	On Demand	Manually
SHA2-224 (A3761)	KAT	CAST	On Demand	Manually
SHA2-224 (A3762)	KAT	CAST	On Demand	Manually
SHA2-256 (A3757)	KAT	CAST	On Demand	Manually
SHA2-256 (A3759)	KAT	CAST	On Demand	Manually
SHA2-256 (A3760)	KAT	CAST	On Demand	Manually
SHA2-256 (A3761)	KAT	CAST	On Demand	Manually
SHA2-256 (A3762)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-256 (A4675)	KAT	CAST	On Demand	Manually
SHA2-256 (A4676)	KAT	CAST	On Demand	Manually
SHA2-384 (A3757)	KAT	CAST	On Demand	Manually
SHA2-384 (A3759)	KAT	CAST	On Demand	Manually
SHA2-384 (A3760)	KAT	CAST	On Demand	Manually
SHA2-384 (A3761)	KAT	CAST	On Demand	Manually
SHA2-384 (A3762)	KAT	CAST	On Demand	Manually
SHA2-512 (A3757)	KAT	CAST	On Demand	Manually
SHA2-512 (A3759)	KAT	CAST	On Demand	Manually
SHA2-512 (A3760)	KAT	CAST	On Demand	Manually
SHA2-512 (A3761)	KAT	CAST	On Demand	Manually
SHA2-512 (A3762)	KAT	CAST	On Demand	Manually
SHA2-512 (A4675)	KAT	CAST	On Demand	Manually
SHA2-512 (A4676)	KAT	CAST	On Demand	Manually
PBKDF (A3757)	KAT	CAST	On Demand	Manually
PBKDF (A3759)	KAT	CAST	On Demand	Manually
PBKDF (A3760)	KAT	CAST	On Demand	Manually
PBKDF (A3761)	KAT	CAST	On Demand	Manually
PBKDF (A3762)	KAT	CAST	On Demand	Manually
PBKDF (A3757)	KAT	CAST	On Demand	Manually
PBKDF (A3759)	KAT	CAST	On Demand	Manually
PBKDF (A3760)	KAT	CAST	On Demand	Manually
PBKDF (A3761)	KAT	CAST	On Demand	Manually
PBKDF (A3762)	KAT	CAST	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3757)	PCT	PCT	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA KeyGen (FIPS186-4) (A3759)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3760)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3761)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A3762)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3757)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3759)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3760)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3761)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A3762)	PCT	PCT	On Demand	Manually

Table 26: Conditional Periodic Information

This information can be found in Section 5.2.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
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.	Failure of pre-operational tests or conditional tests.	The error can be recovered by a restart (i.e., powering off and powering on) of the module.	An error message related to the cause of the failure.
Fatal Error state	The module will abort and will not be available.	Random numbers are requested in the error state or cipher operations	The error can be recovered by a restart (i.e., powering off and	The module is aborted

Name	Description	Conditions	Recovery Method	Indicator
		are requested on a deallocated handle.	powering on) of the module.	

Table 27: Error States

After the pre-operational self-tests and the CASTs succeed, the module becomes operational. If any of the pre-operational self-tests or any of the CASTs fail an error message is returned, and the module transitions to the error state.

When the module fails any pre-operational self-test or conditional test, the module will return an error code to indicate the error and will enter the Error state. Any further cryptographic operation is inhibited. The calling application can obtain the module state by calling the gcry_control(GCRYCTL_OPERATIONAL_P) API function. The function returns FALSE if the module is in the Error state, TRUE if the module is in the Operational state. In the Error state, all data output is inhibited, and no cryptographic operation is allowed. The error can be recovered by a restart (i.e., powering off and powering on) of the module.

If random numbers are requested while the module is in Error state, or if cipher operations are requested on a deallocated handle the module will transition to Fatal Error state, the module will abort and will not be available.

10.5 Operator Initiation of Self-Tests

The software integrity tests and the CASTs can be invoked relying on the gcry_control(GCRYCTL_SELFTEST) API function call or by powering-off and reloading the module. The PCTs can be invoked on demand by requesting the Key Generation service.

10.6 Additional Information

Not Applicable.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Crypto Officer can install the RPM package of the Module (i.e. libgcrypt-1.10.0-10.el9_0.rpm or libgcrypt-1.10.0-10.el9_2.rpm) using standard tools recommended for the installation of RPM packages on a Red Hat Enterprise Linux system (for example, dnf, rpm, and the RHN remote management tool). 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.

Before the RPM package of the module is installed, the RHEL 9 system must operate in FIPS-validated configuration. This can be achieved by:

- Starting the installation in Approved mode. Add the fips=1 option to the kernel command line during the system installation. During the software selection stage, do not install any third-party software.
- Switching the system into Approved mode after the installation. Execute the fips-mode-setup --enable command. Restart the system.

The Crypto Officer must verify the system operates in Approved mode by executing the fips-mode-setup --check command, which should output “FIPS mode is enabled.”

After installation of the RPM package of the module, the operator needs to check the output of the gcry_get_config() API, which should include the following name and version:

Red Hat Enterprise Linux 9 libgcrypt 1.10.0-8b6840b590cedd43

Once libgcrypt has been put into Approved mode, it is not possible to switch back to standard mode without terminating the process first. If the logging verbosity level of libgcrypt has been set to at least 2, the state transitions and the self-tests are logged.

11.2 Administrator Guidance

All the functions, ports and logical interfaces described in this document are available to the Crypto Officer.

The user must not call malloc/free to create/release space for keys, let libgcrypt manage space for keys, which will ensure that the key memory is overwritten before it is released.

gcry_control(GCRYCTL_TERM_SECMEM) needs to be called before the process is terminated.

11.3 Non-Administrator Guidance

The module implements only the Crypto Officer. There are no requirements for non-administrator guidance.

11.4 Design and Rules

Not Applicable.

11.5 Maintenance Requirements

Not Applicable.

11.6 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.

11.7 Additional Information

Not Applicable.

12 Mitigation of Other Attacks

12.1 Attack List

RSA timing attacks.

12.2 Mitigation Effectiveness

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 $y = x\ r^e\ mod\ n$ 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 Mode
DF	Derivation Function
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
KW	AES Key Wrap
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PAI	Processor Algorithm Implementation
PBKDF2	Password-based Key Derivation Function v2
PCT	Pair-wise Consistency Test
PKCS	Public-Key Cryptography Standards
PR	Prediction Resistance
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSP	Sensitive Security Parameter
XOF	Extendable Output Function
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

FIPS140-3	**FIPS PUB 140-3 - Security Requirements For Cryptographic Modules** March 2019 https://doi.org/10.6028/NIST.FIPS.140-3
FIPS140-3_IG	**Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program** March 2024 https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf
FIPS140-3_MM	**FIPS 140-3 Cryptographic Module Validation Program - Management Manual (Draft)** December 2022 https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/Draft%20FIPS-140-3-CMVP%20Management%20Manual%20v1.2%20%5BDec%2023%202022%5D.pdf
FIPS180-4	**Secure Hash Standard (SHS)** August 2015 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS186-4	**Digital Signature Standard (DSS)** July 2013 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS186-2	**Digital Signature Standard (DSS)** January 2000 https://csrc.nist.gov/files/pubs/fips/186-2/final/docs/fips186-2.pdf
FIPS197	**Advanced Encryption Standard** November 2001 https://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
FIPS198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography** Specifications Version 2.1 February 2003 https://www.ietf.org/rfc/rfc3447.txt
SP800-38A	**NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf
SP800-38B	**NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication** May 2005 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38b.pdf
SP800-38C	**NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality** May 2004 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
SP800-38E	**NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices** January 2010 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf

SP800-38F	**NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP800-90Arev1	**NIST Special Publication 800-90A Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP800-90B	**NIST Special Publication 800-90B - Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf
SP800-132	**NIST Special Publication 800-132 - Recommendation for Password-Based Key Derivation - Part 1: Storage Applications** December 2010 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf
SP800-133rev2	**NIST Special Publication 800-133 - Recommendation for Cryptographic Key Generation** June 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf
SP800-140Br1	**NIST Special Publication 800-140B – Revision 1 - CMVP Security Policy Requirements** November 2023 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140Br1.pdf