# Canonical Ltd.

# Canonical Ltd. Ubuntu 24.04 GnuTLS Cryptographic Module

# Version 3.8.3-1.1ubuntu3.1+Fips1

# FIPS 140-3 Non-Proprietary Security Policy

## Version 1.1

## Last updated: 2026-02-17


**Prepared by:**
atsec information security corporation
4516 Seton Center Parkway, Suite 250
Austin, TX 78759
www.atsec.com

**Prepared for:**
Canonical Ltd.
110 Southwark Street, Blue Fin Building, 5th Floor
London, SE1 0SU
www.canonical.com

# Table of Contents



© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice. 

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice. 

# List of Tables



# List of Figures



© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice. 

# 1 General

## 1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for version 3.8.3-1.1ubuntu3.1+Fips1 of the Canonical Ltd. Ubuntu 24.04 GnuTLS Cryptographic Module. It has a one-to-one mapping to SP 800-140B starting with section B.2.1 named “General” that maps to section 1 in this document and ending with section B.2.12 named “Mitigation of other attacks” that maps to section 12 in this document.

## 1.2 Security Levels

| Section | Title | Security Level |
|---|---|---|
| 1 | General | 1 |
| 2 | Cryptographic module specification | 1 |
| 3 | Cryptographic module interfaces | 1 |
| 4 | Roles, services, and authentication | 1 |
| 5 | Software/Firmware security | 1 |
| 6 | Operational environment | 1 |
| 7 | Physical security | N/A |
| 8 | Non-invasive security | N/A |
| 9 | Sensitive security parameter management | 1 |
| 10 | Self-tests | 1 |
| 11 | Life-cycle assurance | 1 |
| 12 | Mitigation of other attacks | N/A |
| | Overall Level | 1 |

Table 1: Security Levels

## 1.3 Additional Information

N/A

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 2 Cryptographic Module Specification

## 2.1 Description

**Purpose and Use:**

The Canonical Ltd. Ubuntu 24.04 GnuTLS Cryptographic Module (hereafter referred to as “the module”) provides cryptographic services to applications running in the user space of the underlying operating system through a C language Application Program Interface (API).

**Module Type**: Software

**Module Embodiment**: MultiChipStand

**Cryptographic Boundary:**

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

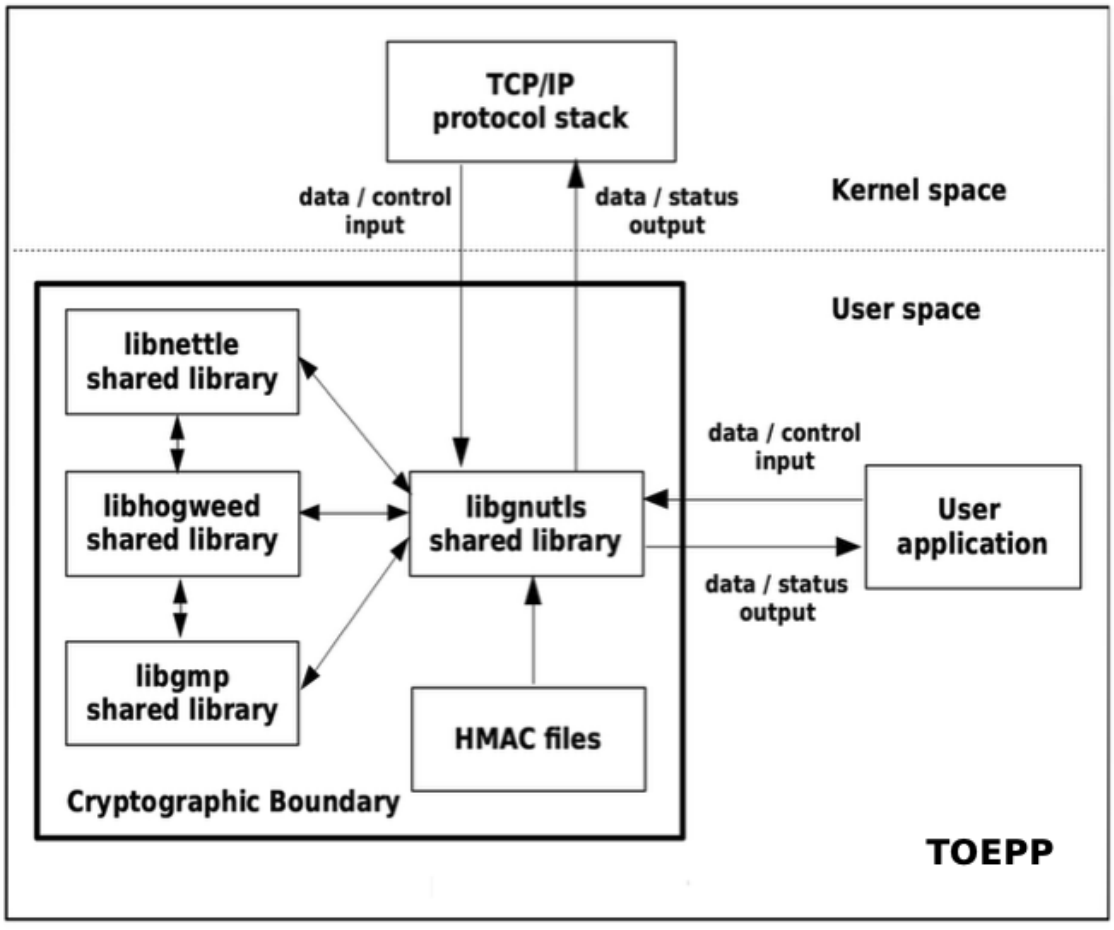


Figure 1: Block Diagram

**Tested Operational Environment’s Physical Perimeter (TOEPP):**

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

The TOEPP (tested operational environment's physical perimeter) of the module is defined as the general-purpose computer on which the module is installed.

## 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 |
|---|---|---|---|
| libgnutls.so.30, libnettle.so.8, libhogweed.so.6, libgmp.so.10 on Ubuntu 24.04 with Intel Xeon Gold 6226 | 3.8.3-1.1ubuntu3.1+Fips1 | N/A | HMAC-SHA2-256 |
| libgnutls.so.30, libnettle.so.8, libhogweed.so.6, libgmp.so.10 on Ubuntu 24.04 with AWS Graviton3 | 3.8.3-1.1ubuntu3.1+Fips1 | N/A | HMAC-SHA2-256 |
| libgnutls.so.30, libnettle.so.8, libhogweed.so.6, libgmp.so.10 on Ubuntu 24.04 with IBM Telum | 3.8.3-1.1ubuntu3.1+Fips1 | N/A | HMAC-SHA2-256 |

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) |
|---|---|---|---|---|---|
| Ubuntu 24.04 | Supermicro SYS-1019P-WTR | Intel Xeon Gold 6226 | Yes | N/A | 3.8.3-1.1ubuntu3.1+Fips1 |
| Ubuntu 24.04 | Amazon Web Services (AWS) c7g.metal | AWS Graviton3 | Yes | N/A | 3.8.3-1.1ubuntu3.1+Fips1 |
| Ubuntu 24.04 | IBM z16 | IBM Telum | Yes | N/A | 3.8.3-1.1ubuntu3.1+Fips1 |
| Ubuntu 24.04 | Supermicro SYS-1019P-WTR | Intel Xeon Gold 6226 | No | N/A | 3.8.3-1.1ubuntu3.1+Fips1 |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Operating System | Hardware Platform | Processors | PAA/PAI | Hypervisor or Host OS | Version(s) |
|---|---|---|---|---|---|
| Ubuntu 24.04 | Amazon Web Services (AWS) c7g.metal | AWS Graviton3 | No | N/A | 3.8.3-1.1ubuntu3.1+Fips1 |
| Ubuntu 24.04 | IBM z16 | IBM Telum | No | N/A | 3.8.3-1.1ubuntu3.1+Fips1 |

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

The module makes use of hardware acceleration provided by the hardware platform. Namely, AES-NI from the Intel based platform, NEON and Cryptography Extension (CE) for the Graviton3 based platform and CPACF for the IBM Telum based platform, listed in the *Tested Operational Environments - Software, Firmware, Hybrid* table. Out of these, only CPACF is considered as PAI and other two are considered as PAA.

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

N/A for this module.

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

N/A

## 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 | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() |
| Non-approved mode | Automatically entered whenever a non-approved service is requested | Non-Approved | Return GNUTLS_FIPS140_OP_NOT_APPROVED from the function: gnutls_fips140_get_operation_state() |

Table 4: Modes List and Description

When the module starts up successfully, after passing all the pre-operational and conditional cryptographic algorithms self-tests (CASTs), the module is operating in the approved mode of operation by default. Please see section 4 for the details on service indicator provided by the module that identifies when an approved service is called.

**Mode Change Instructions and Status:**

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

If the module is in the approved mode, it can be transitioned to the non-approved mode by calling one of the non-approved services listed in section 4. If the module is in the non-approved mode, the module can be transitioned to the approved mode by calling one of the approved services listed in section 4.

**Degraded Mode Description:**

N/A

## 2.5 Algorithms

**Approved Algorithms:**

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| AES-CBC | A5671, A5672, A5673, A5674, A5679, A5713, A5935, A5936, A5939 | - | SP 800-38A |
| AES-CCM | A5671, A5713, A5935 | - | SP 800-38C |
| AES-CFB8 | A5676, A5677, A5682, A5942 | - | SP 800-38A |
| AES-CMAC | A5671, A5674, A5679, A5935, A5939 | - | SP 800-38B |
| AES-GCM | A5671, A5672, A5673, A5674, A5679, A5713, A5935, A5936, A5939 | - | SP 800-38D |
| AES-GMAC | A5679, A5939 | - | SP 800-38D |
| AES-XTS Testing Revision 2.0 | A5680, A5940 | - | SP 800-38E |
| Counter DRBG | A5679, A5939 | - | SP 800-90A Rev. 1 |
| ECDSA KeyGen (FIPS186-5) | A5679, A5939 | - | FIPS 186-5 |
| ECDSA KeyVer (FIPS186-5) | A5679, A5939 | - | FIPS 186-5 |
| ECDSA SigGen (FIPS186-5) | A5679, A5939 | - | FIPS 186-5 |
| ECDSA SigVer (FIPS186-4) | A5679, A5939 | - | FIPS 186-4 |
| ECDSA SigVer (FIPS186-5) | A5679, A5939 | - | FIPS 186-5 |
| HMAC-SHA-1 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 198-1 |
| HMAC-SHA2-224 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 198-1 |
| HMAC-SHA2-256 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 198-1 |
| HMAC-SHA2-384 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 198-1 |
| HMAC-SHA2-512 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 198-1 |
| KAS-ECC-SSC Sp800-56Ar3 | A5679, A5939 | - | SP 800-56A Rev. 3 |
| KAS-FFC-SSC Sp800-56Ar3 | A5679, A5939 | - | SP 800-56A Rev. 3 |
| KDA HKDF Sp800-56Cr1 | A5678, A5938 | - | SP 800-56C Rev. 2 |
| KDF TLS (CVL) | A5679, A5939 | - | SP 800-135 Rev. 1 |
| PBKDF | A5679, A5939 | - | SP 800-132 |
| RSA KeyGen (FIPS186-5) | A5679, A5939 | - | FIPS 186-5 |
| RSA SigGen (FIPS186-5) | A5679, A5939 | - | FIPS 186-5 |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| RSA SigVer (FIPS186-2) | A5679, A5939 | - | FIPS 186-4 |
| RSA SigVer (FIPS186-4) | A5679, A5939 | - | FIPS 186-4 |
| RSA SigVer (FIPS186-5) | A5679, A5939 | - | FIPS 186-5 |
| Safe Primes Key Generation | A5679, A5939 | - | SP 800-56A Rev. 3 |
| SHA-1 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 180-4 |
| SHA2-224 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 180-4 |
| SHA2-256 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 180-4 |
| SHA2-384 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 180-4 |
| SHA2-512 | A5674, A5679, A5713, A5937, A5939 | - | FIPS 180-4 |
| SHA3-224 | A5675, A5681, A5941 | - | FIPS 202 |
| SHA3-256 | A5675, A5681, A5941 | - | FIPS 202 |
| SHA3-384 | A5675, A5681, A5941 | - | FIPS 202 |
| SHA3-512 | A5675, A5681, A5941 | - | FIPS 202 |
| TLS v1.2 KDF RFC7627 (CVL) | A5679, A5939 | - | SP 800-135 Rev. 1 |

Table 5: Approved Algorithms

**Vendor-Affirmed Algorithms:**

| Name | Properties | Implementation | Reference |
|---|---|---|---|
| Cryptographic Key Generation (CKG) | Key Type:Symmetric and Asymmetric | N/A | SP 800-133r2 section 4 example 1 without the use of V (refer to additional comment 2 of IG D.H) |

Table 6: Vendor-Affirmed Algorithms

**Non-Approved, Allowed Algorithms:**

N/A for this module.

**Non-Approved, Allowed Algorithms with No Security Claimed:**

| Name | Caveat | Use and Function |
|---|---|---|
| MD5 | Only allowed as the PRF in TLSv1.0 and v1.1 per IG 2.4.A | Message digest used in TLS 1.0 / 1.1 KDF only for legacy use |

Table 7: Non-Approved, Allowed Algorithms with No Security Claimed

**Non-Approved, Not Allowed Algorithms:**

| Name | Use and Function |
|---|---|
| Camellia | Symmetric encryption; Symmetric decryption |
| ChaCha20 | Symmetric encryption; Symmetric decryption |
| Chacha20 and Poly1305 | Authenticated encryption; Authenticated decryption |
| CMAC with Triple-DES | Message authentication code (MAC) |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Use and Function |
|---|---|
| DES | Symmetric encryption; Symmetric decryption |
| ECDSA (with curves other than P-256, P-384, P-512) | Key generation; Public key verification |
| ECDSA (with curves other than P-256, P-384, P-512 or hash functions other than SHA2-224, SHA2-256, SHA2-384, SHA2-512) | Digital signature generation; Digital signature verification |
| EC Diffie-Hellman (with curves other than P-256, P-384, P-512) | Key agreement; Shared secret computation |
| GMAC | Message authentication code (MAC) |
| GOST | Symmetric encryption; Symmetric decryption; Message digest |
| HMAC (with keys smaller than 112-bits) | Message authentication code (MAC) |
| HMAC (with GOST) | Message authentication code (MAC) |
| MD2, MD5 | Message digest; Message authentication code (MAC) |
| PBKDF (with non-approved message digest algorithms or using input parameters not meeting requirements stated in section 2.7 of the security policy) | Key derivation |
| RC2, RC4 | Symmetric encryption; Symmetric decryption |
| RMD160 | Message digest; Message authentication code (MAC) |
| RSA (with keys smaller than 2048 bits and/or hash functions other than SHA2-224, SHA2-256, SHA2-384, SHA2-512) | Digital signature generation |
| RSA (with keys smaller than 1024 bits and/or hash functions other than SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512) | Digital signature verification |
| RSA (with any key sizes) | Key encapsulation; Key un-encapsulation |
| Salsa20 | Symmetric encryption; Symmetric decryption |
| SRP | Key agreement; Shared secret computation |
| STREEBOG | Message digest; Message authentication code (MAC) |
| Triple-DES | Symmetric encryption; Symmetric decryption |
| UMAC | Message authentication code (MAC) |
| Yarrow | Random number generation |
| AES-GCM (when not used in the context of the TLS protocol) | Authenticated encryption; Authenticated decryption |
| DSA | Key generation; Domain parameter generation; Digital signature generation; Digital signature verification |
| Diffie-Hellman (with domain parameters other than safe primes) | Key agreement; Shared secret computation |
| Symmetric Keygen (with keys smaller than 112 bits) | Key generation |

Table 8: Non-Approved, Not Allowed Algorithms

## 2.6 Security Function Implementations

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| Symmetric encryption | BC-UnAuth | Symmetric encryption | | AES-CBC: (A5671, A5672, A5673, A5674, A5679, A5713, A5935, A5936, A5939)<br>AES-CFB8: (A5676, A5677, A5682, A5942)<br>AES-XTS Testing Revision 2.0: (A5680) |
| Symmetric decryption | BC-UnAuth | Symmetric decryption | | AES-CBC: (A5671, A5672, A5673, A5674, A5679, A5713, A5935, A5936)<br>AES-CFB8: (A5676, A5677, A5682)<br>AES-XTS Testing Revision 2.0: (A5940) |
| Message authentication code (MAC) | MAC | Message authentication code (MAC) | | HMAC-SHA-1: (A5674, A5679, A5713, A5937, A5939)<br>HMAC-SHA2-224: (A5674, A5679, A5713, A5937, A5939)<br>HMAC-SHA2-256: (A5674, A5679, A5713, A5939, A5937)<br>HMAC-SHA2-384: (A5674, A5679, A5713, A5937, A5939)<br>HMAC-SHA2-512: (A5674, A5679, A5713, A5937, A5939)<br>AES-CMAC: (A5935, A5671, A5674, A5679, A5939)<br>AES-GMAC: (A5679, A5939) |
| Message digest | SHA | Message digest | | SHA-1: (A5674, A5679, A5713, A5937, A5939)<br>SHA2-224: (A5674, A5679, A5713, A5937, A5939) |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | | | SHA2-256: (A5674, A5679, A5713, A5937, A5939)<br>SHA2-384: (A5674, A5679, A5713, A5937, A5939)<br>SHA2-512: (A5674, A5679, A5713, A5937, A5939)<br>SHA3-224: (A5675, A5681, A5941)<br>SHA3-256: (A5675, A5681, A5941)<br>SHA3-384: (A5675, A5681, A5941)<br>SHA3-512: (A5675, A5681, A5941) |
| Deterministic random bit generation | DRBG | Deterministic random bit generation | | Counter DRBG: (A5679, A5939) |
| Asymmetric key generation | AsymKeyPair-KeyGen<br>CKG | Asymmetric key generation | | ECDSA KeyGen (FIPS186-5): (A5679, A5939)<br>RSA KeyGen (FIPS186-5): (A5679, A5939)<br>Safe Primes Key Generation: (A5679, A5939) |
| Public key verification | AsymKeyPair-KeyVer | Public key verification | | ECDSA KeyVer (FIPS186-5): (A5679, A5939) |
| Digital signature generation | DigSig-SigGen | Digital signature generation | IG:IG C.F. The module supports RSA modulus sizes which are not tested by CAVP. | ECDSA SigGen (FIPS186-5): (A5679, A5939)<br>RSA SigGen (FIPS186-5): (A5679, A5939) |
| Digital signature verification | DigSig-SigVer | Digital signature verification | IG:IG C.F. The module supports RSA modulus sizes which are not tested by CAVP. | ECDSA SigVer (FIPS186-5): (A5679, A5939)<br>RSA SigVer (FIPS186-2): (A5679, A5939)<br>RSA SigVer (FIPS186-5): (A5679, A5939) |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| (Diffie-Hellman) shared secret computation | KAS-SSC | Diffie-Hellman shared secret computation | IG:IG D.F scenario 2 path (1) | KAS-FFC-SSC Sp800-56Ar3: (A5679, A5939)<br>KAS-ECC-SSC Sp800-56Ar3: (A5679, A5939) |
| Key derivation | KAS-135KDF<br>KAS-56CKDF<br>PBKDF | Key derivation | | KDF TLS: (A5679, A5939)<br>PBKDF: (A5679, A5939)<br>TLS v1.2 KDF RFC7627: (A5679, A5939)<br>KDA HKDF Sp800-56Cr1: (A5678, A5938) |
| (EC Diffie-Hellman) shared secret computation | KAS-SSC | EC Diffie-Hellman shared secret computation | IG: IG D.F scenario 2 path (1) | KAS-ECC-SSC Sp800-56Ar3: (A5679, A5939) |
| KAS1 (EC Diffie-Hellman) | KAS-Full | EC Diffie-Hellman key agreement | IG:IG D.F scenario 2 path (2) | KAS-ECC-SSC Sp800-56Ar3: (A5679, A5939)<br>KDF TLS: (A5679, A5939)<br>TLS v1.2 KDF RFC7627: (A5679, A5939)<br>KDA HKDF Sp800-56Cr1: (A5678, A5938) |
| KAS2 (Diffie-Hellman) | KAS-Full | Diffie-Hellman key agreement | IG:IG D.F scenario 2 path (2) | KAS-FFC-SSC Sp800-56Ar3: (A5679, A5939)<br>KDF TLS: (A5679, A5939)<br>TLS v1.2 KDF RFC7627: (A5679, A5939)<br>KDA HKDF Sp800-56Cr1: (A5678, A5938) |
| Authenticated encryption | BC-Auth | Authenticated symmetric encryption. AES-GCM is considered approved by the module only used in the context of the TLS protocol. | | AES-CCM: (A5671, A5713, A5935)<br>AES-GCM: (A5671, A5672, A5673, A5674, A5679, A5713, A5935, A5936, A5939) |
| Authenticated decryption | BC-Auth | Authenticated symmetric decryption. AES-GCM is | | AES-CCM: (A5671, A5713, A5935)<br>AES-GCM: (A5671, A5672, A5673, |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | considered approved by the module only used in the context of the TLS protocol. | | A5674, A5679, A5713, A5935, A5936, A5939) |
| Symmetric Key Generation | CKG | Symmetric Key Generation | | Counter DRBG: (A5679, A5939) |
| Digital Signature Verification (Legacy) | DigSig-SigVer | Legacy digital signature verification | Reference :FIPS186-2, FIPS186-4 IG:IG C.M | ECDSA SigVer (FIPS186-4): (A5679, A5939) RSA SigVer (FIPS186-2): (A5679, A5939) RSA SigVer (FIPS186-4): (A5679, A5939) |

Table 9: Security Function Implementations

## 2.7 Algorithm Specific Information

**Hash Algorithms**

In compliance with IG C.B, every approved hash algorithm implementation was CAVP tested and validated on all the module's operational environments. Section 2.5 of this security policy contains a table of the CAVP certificates of the approved hash functions.

For the higher-level algorithms that use the approved hash functions - Counter DRBG, ECDSA SigGen, ECDSA SigVer, HMAC, KDA HKDF Sp800-56Cr1, KDF TLS (CVL), TLS v1.2 KDF RFC7627 (CVL), PBKDF2, RSA SigGen, RSA SigVer – every implemented combination for which CAVP testing exists was CAVP tested and validated on all the module's operational environments. Section 2.5 of this security policy contains a table of the CAVP certificates of these higher-level algorithms.

**SHA-3**

The module provides SHA-3 hash functions compliant with IG C.C. Every implementation of each SHA-3 function was tested and validated on all the module's operating environments. SHAKE functions are not implemented. SHA-3 hash functions are not used as part of a higher-level algorithm.

**RSA Key Generation**

In compliance with IG C.E, the module generates RSA signature keys using an approved method of FIPS 186-5: generation of random primes that are provably prime. The CAVP certificate in table 2.5 indicates that the RSA key generating algorithm has been tested and validated for conformance to the methods in FIPS 186-5.

**RSA Signature Generation and Signature Verification**

The module provides RSA signature generation and signature verification compliant with IG C.F. The module supports RSA modulus lengths greater than or equal to 2048 bits for both signature generation and signature verification. The RSA signature generation and signature verification implementations have been tested for all module lengths available in CAVP testing: 2048, 3072, and 4096 bits. Legacy 186-2 signature verification is supported for modulus sizes of 1024, 1280, 1536, and 1792 bits. Legacy 186-4 signature verification is supported for the modulus size 1024. The RSA signature verification has been

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

tested for all legacy module lengths available for CAVP testing: 1024 and 1530 bits. The number of Miller-Rabin tests is consistent with the bit sizes of p and q from Table B.1 of FIPS 186-5.

**AES-GCM**

The module implements AES GCM for being used in the TLS v1.2 and v1.3 protocols. AES GCM IV generation is compliant with [FIPS140-3_IG] IG C.H for both protocols as follows:

- For TLS v1.2, IV generation is compliant with scenario 1.a of IG C.H and [RFC5288]. The module supports acceptable AES-GCM cipher suites from section 3.3.1 of [SP800-52rev2].

- For TLS v1.3, IV generation is compliant with scenario 5 of IG C.H and [RFC8446]. The module supports acceptable AES-GCM cipher suites from section 3.3.1 of [SP800-52rev2].

The IV generated in both scenarios is only used within the context of the TLS protocol implementation. The nonce explicit part of the IV does not exhaust the maximum number of possible values for a given session key. The design of the TLS protocol in this module implicitly ensures that the nonce explicit, or counter portion of the IV will not exhaust all its possible values.

In case the module's power is lost and then restored, the key used for the AES GCM encryption or decryption shall be redistributed.

**AES-XTS**

The AES algorithm in XTS mode can only be used for the cryptographic protection of data on storage devices, as specified in SP 800-38E. The length of a single data unit encrypted with the AES-XTS 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. Key_1 and Key_2 shall be generated and/or established independently according to the rules for component symmetric keys from NIST SP 800-133r2, Section 6.3

**Key Agreement Methods**

To comply with the assurances found in Section 5.6.2 of SP 800-56A Rev. 3, the operator must use the module together with an application that implements the TLS protocol. Additionally, the module's approved "Asymmetric key generation" service must be used to generate ephemeral Diffie-Hellman or EC Diffie-Hellman key pairs, or the key pairs must be obtained from another FIPS-validated module.

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

**Key Transport Methods**

Please refer to section 2.10 of this security policy.

**Cryptographic Key Generation**

In compliance with IG D.H, the module generates symmetric keys and seeds for asymmetric keys using the method described in section 4 example 1 of SP 800-133 Rev. 2 without the use of V (direct DRBG output as described in additional comment 2 of IG D.H).

Please refer to section 2.9 for more information on the key generation methods employed by the module.

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice. 

**DRBGs**

In compliance with IG D.L, the entropy input, DRBG seed, DRBG internal state (values of V and Key) are considered CSPs.

The DRBG internal state is contained within the DRBG mechanism boundary and is not accessible by other mechanisms.

**PBKDF2**

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

In accordance with SP 800-132, 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 length of the MK or DPK shall be 112 bits or more (this is verified by the module to determine the service is approved).

- A portion of the salt, with a length of at least 128 bits (this is verified by the module to determine the service is approved), shall be generated randomly using the SP 800-90A Rev. 1 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 shall be 1000 (this is verified by the module to determine the service is approved).

- Passwords or passphrases, used as an input for the PBKDF, shall not be used as cryptographic keys.

- The length of the password or passphrase shall be of at least 8 characters (this is verified by the module to determine the service is approved), and shall consist of lower-case, upper-case, and numeric characters. The probability of guessing the value is estimated to be 1/62^{8} = 10^{-14}, which is less than 2^{-112}. If the password consists of only digits (worst case), the probability of guessing the value is estimated to be 10^{-8} which is less than 2^{-112}.

The requirements of input parameters (derived key length, salt length, iteration count and password length) are verified by the module when providing the service indicator.

The calling application shall also observe the rest of the requirements and recommendations specified in SP 800-132.

**TLS v1.2 KDF**

In compliance with IG D.Q, the module supports the TLS 1.2 KDF with the extended master secret: TLS v1.2 KDF RFC7627 (CVL).

**SHA-1**

SHA-1 is only approved when used for Message Digest, Message Authentication Code, and Digital Signature Verification (Legacy). The use of SHA-1 for Digital Signature Generation is non-approved.

**Legacy Algorithms**

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice. 

The cryptographic module implements the following cryptographic algorithms for legacy use. Algorithms designated as "Legacy" can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M:

- RSA SigVer (FIPS 186-4) with 1024-bit keys or SHA-1
- RSA SigVer (FIPS 186-2) with 1536-bit keys or SHA-1
- ECDSA SigVer (FIPS-186-4) with SHA-1.

## 2.8 RBG and Entropy

| Cert Number | Vendor Name |
|---|---|
| E217 | Canonical |

Table 10: Entropy Certificates

| Name | Type | Operational Environment | Sample Size | Entropy per Sample | Conditioning Component |
|---|---|---|---|---|---|
| Canonical Userspace CPU Time Jitter Entropy source | Non-Physical | Ubuntu 24.04 on Supermicro SYS-1019P-WTR on Intel Xeon Gold 6226; Ubuntu 24.04 on Amazon Web Services (AWS) c7g.metal on AWS Graviton3; Ubuntu 24.04 on IBM Telum on IBM z16 | 256 | 256 | SHA3-256 cert. A5588; AES-256 CTR DRBG cert. A5588, A5591, A5592, A5593, A5594, A5595, A5599, A5600, A5601, A5602, A5603, A5606, A5607, A5608, A5609, A5610, A5611 |

Table 11: Entropy Sources

The module implements CTR_DRBG with AES-256, according to SP 800-90A Rev. 1, without a derivation function and without prediction resistance. The module uses an SP 800-90B-compliant entropy source as specified above. This entropy source is located within the physical perimeter, but outside of the cryptographic boundary of the module.

The module obtains 384 bits from the entropy source to seed the DRBG, and this is sufficient to provide a DRBG with 256 bits of security strength. The largest key strength generated by the module is 256 bits.

## 2.9 Key Generation

The module generates symmetric keys and seeds for asymmetric keys using the method described in section 4 example 1 of SP 800-133 Rev. 2 without the use of V (direct output of DRBG as described in additional comment 2 of IG D.H).

The module generates the following keys:

- RSA (asymmetric): 2048, 3072, 4096, 6144, 8192-bit keys with 112-200 bits of key strength
- ECDSA (asymmetric): P-256, P-384, P-521 elliptic curves with 128-256 bits of key strength
- ECDH (asymmetric): P-256, P-384, P-521 elliptic curves with 128-256 bits of key strength
- Safe Primes (asymmetric): 2048, 3072, 4096, 6144, 8192-bit keys with 112-200 bits of key strength
- AES (symmetric): 128, 192, 256-bit keys with 128-256 bits of key strength
- HMAC (symmetric): 112-524288 bit keys with 112-256 bits of key strength

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

The generation of RSA keys is compliant with section A.1.2 of FIPS 186-5 (provable primes).

The generation of ECDSA keys is compliant with section A.2.2 of FIPS186-5 (testing candidates).

The generation of EC Diffie-Hellman keys is performed using the ECDSA key generation method, which is compliant with FIPS 186-5 and SP 800-56A Rev. 3.

The generation of Diffie-Hellman keys is compliant with SP 800-56A Rev. 3 (testing candidates). The module generates keys using safe primes defined in RFC7919 and RFC3526.

Additionally, the module implements the following key derivation methods:

- KDF TLS (CVL), compliant with SP800-135 Rev. 1: derivation of secret keys in the context of TLS 1.0/1.1
- TLS v1.2 KDF RFC7627 (CVL), compliant with SP 800-135 Rev. 1: derivation of secret keys in the context of TLS 1.2
- KDA HKDF SP 800-56Cr1, compliant with SP 800-56C Rev. 1: derivation of secret keys in the context of SP 800-56A Rev. 3 key agreement schemes
- PBKDF2, compliant with option 1a of SP 800-132: derivation of keys for use in storage applications

## 2.10 Key Establishment

**Key Agreement**

The module provides Diffie-Hellman and EC Diffie-Hellman shared secret computation compliant with SP 800-56A Rev. 3, in accordance with scenario 2 (1) of IG D.F and used as part of the TLS protocol key exchange in accordance with scenario 2 (2) of IG D.F; that is, the shared secret computation (KAS-FFC-SSC and KAS-ECC-SSC) followed by the derivation of the keying material using KDF TLS (CVL), TLS v1.2 KDF RFC7627 (CVL), or KDA HKDF SP 800-56C Rev. 1. The Diffie-Hellman shared secret computation, EC Diffie-Hellman shared secret computation, KDF TLS (CVL), TLS v1.2 KDF RFC7627 (CVL), and KDA HKDF SP 800-56C Rev. 1 have been CAVP tested.

The EC Diffie-Hellman shared secret computation uses the Ephemeral Unified Model. The module supports EC Diffie-Hellman shared secret computation with P-256, P-384, and P-521 curves which have a security strength of 128-256 bits.

The Diffie-Hellman shared secret computation uses the DH Ephemeral scheme. The module supports Diffie-Hellman shared secret computation with the MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, and ffdhe8192 groups which have a security strength of 112-200 bits.

**Key Transport**

The module provides key wrapping (KTS), compliant with IG D.G, using AES-CCM, AES-GCM, and AES-CBC with HMAC, used in the context of the TLS protocol cipher suites with 128-bit or 256-bit keys, with strengths of 128 bits and 256 bits respectively. When using AES-CBC with HMAC, the entire wrapped message is authenticated. AES-CCM, AES-GCM, AES-GCM, and HMAC have been tested and validated by the CAVP and the corresponding certificate numbers are in section 2.5 of the security policy.

## 2.11 Industry Protocols

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

The TLS protocol implementation provides both server and client sides. To operate in the approved mode, digital certificates used for server and client authentication shall comply with the restrictions of key size and message digest algorithms imposed by SP 800-131A Rev. 2.

No parts of the TLS protocol, other than the approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP.

## 2.12 Additional Information

N/A

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 3 Cryptographic Module Interfaces

## 3.1 Ports and Interfaces

| Physical Port | Logical Interface(s) | Data That Passes |
|---|---|---|
| N/A | Data Input | API input parameters for data. |
| N/A | Data Output | API output parameters for data. |
| N/A | Control Input | API function calls. |
| N/A | Status Output | API return codes, error message. |

Table 12: Ports and Interfaces

The module does not have a control output interface.

## 3.2 Trusted Channel Specification

N/A

## 3.3 Control Interface Not Inhibited

N/A

## 3.4 Additional Information

N/A

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 4 Roles, Services, and Authentication

## 4.1 Authentication Methods

N/A for this module.

## 4.2 Roles

| Name | Type | Operator Type | Authentication Methods |
|---|---|---|---|
| Crypto Officer | Role | CO | None |

Table 13: Roles

## 4.3 Approved Services

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| Encryption | Encryption | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | AES key, plaintext | Ciphertext | Key derivation | Crypto Officer - AES key: W,E |
| Decryption | Decryption | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | AES key, ciphertext | Plaintext | Symmetric decryption Authenticated decryption | Crypto Officer - AES key: W,E |
| Authenticated Encryption | Authenticated encryption | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | AES key, plaintext, IV | Ciphertext, MAC tag | Authenticated encryption | Crypto Officer - AES key: W,E |
| Authenticated Decryption | Authenticated decryption | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | AES key, ciphertext, IV, MAC tag | Plaintext or fail | Authenticated decryption | Crypto Officer - AES key: W,E |
| Message Authentication Generation | MAC computation | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | AES key or HMAC key, message | MAC tag | Message authentication code (MAC) | Crypto Officer - HMAC key: W,E - AES key: W,E |
| Message Digest | Generating message digest | Return GNUTLS_FIPS140_OP_A | Message | Message digest | Message digest | Crypto Officer |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | PPROVED from the function: gnutls_fips140_get_operation_state() | | | | |
| Random Number Generation | Generating random numbers | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | Output length | Random bytes | Deterministic random bit generation | Crypto Officer - Entropy input: W,E - DRBG seed: G,E - DRBG internal state: G,W,E |
| Asymmetric Key Generation | Generates a key pair | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | Modulus size / Curve or key size | Module Generated RSA public key, Module Generated RSA private key / Module Generated EC public key, Module Generated EC private key / Module Generated ECDSA public key, Module Generated ECDSA private key | Asymmetric key generation | Crypto Officer - Module Generated RSA public key: G,R - Module Generated RSA private key: G,R - Module Generated ECDSA public key: G,R - Module Generated ECDSA private key: G,R - Module Generated EC Diffie-Hellman public key: G,R |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - Module Generated EC Diffie-Hellman private key: G,R<br>- Module Generated Diffie-Hellman public key: G,R<br>- Module Generated Diffie-Hellman private key: G,R |
| Symmetric Key Generation | Generate symmetric key | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | Key size | Module Generated AES key / Module Generated HMAC key | Symmetric Key Generation | Crypto Officer<br>- Module Generated AES key: G,R<br>- Module Generated HMAC key: G,R |
| Key Verification | Verifying the public key | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | Public key | Success/ error | Public key verification | Crypto Officer<br>- ECDSA public key: W,E<br>- EC Diffie-Hellman public key from peer: W,E |
| Signature Generation | Generating signature | Return GNUTLS_FIPS140_OP_APPROVED from the | Message, ECDSA | Digital signature | Digital signature | Crypto Officer<br>- RSA |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | function: gnutls_fips140_get_operation_state() | private key or RSA private key | | generation | private key: W,E<br>- ECDSA private key: W,E |
| Signature Verification | Verifying signature | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | Signature, ECDSA public key or RSA public key | Digital signature verification result | Digital signature verification | Crypto Officer<br>- RSA public key: W,E<br>- ECDSA public key: W,E |
| Shared Secret Computation | Calculating the Shared Secret | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | Public key from peer, private key | Shared secret | (EC Diffie-Hellman) shared secret computation (Diffie-Hellman) shared secret computation | Crypto Officer<br>- Diffie-Hellman public key from peer: W,E<br>- Diffie-Hellman private key: W,E<br>- EC Diffie-Hellman public key from peer: W,E<br>- EC Diffie-Hellman private key: W,E<br>- Diffie-Hellman Shared secret: G,R<br>- EC Diffie-Hellman Shared secret: G,R |
| Key Derivatio | Deriving Keys | Return GNUTLS_FIPS140_OP_APPROVED from the | TLS pre- | TLS derived key | Key derivation | Crypto Officer<br>- TLS |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| n with TLS KDF | | function: gnutls_fips140_get_operation_state() | master secret | | | pre-master secret: W,E<br>- TLS master secret: G,E<br>- TLS derived key: G,R |
| Key Derivation with PBKDF | Deriving Keys | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | Password, salt, iteration count | PBKDF derived key | Key derivation | Crypto Officer<br>- PBKDF password or passphrase: W,E<br>- PBKDF derived key: G,R |
| Key Derivation with KDA HKDF | Deriving Keys | Return GNUTLS_FIPS140_OP_APPROVED from the function: gnutls_fips140_get_operation_state() | Shared secret, key length, digest | HKDF derived key | Key derivation | Crypto Officer<br>- HKDF derived key: G,R<br>- EC Diffie-Hellman Shared secret: W,E<br>- Diffie-Hellman Shared secret: W,E |
| Transport Layer Security (TLS) Network Protocol | Provide supported cipher suites (listed in Appendix A) in approved mode | GNUTLS_FIPS140_OP_APPROVED | Cipher-suites listed in Appendix A, Digital Certificate, Public and Private Keys, Application Data | codes and/or log messages, Application data | Symmetric encryption Symmetric decryption Message authentication code (MAC) Message digest Asymmet | Crypto Officer<br>- RSA public key: W,E<br>- RSA private key: W,E<br>- ECDSA public key: W,E<br>- ECDSA private key: W,E<br>- TLS pre- |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | ric key generation Public key verification Digital signature generation Digital signature verification KAS1 (EC Diffie-Hellman) KAS2 (Diffie-Hellman) Authenticated encryption Authenticated decryption Digital Signature Verification (Legacy) | master secret: G,E - TLS master secret: G,E,Z - Module Generated ECDSA public key: G,E - Module Generated ECDSA private key: G,E - Module Generated EC Diffie-Hellman public key: G,E - Module Generated EC Diffie-Hellman private key: G,E - Diffie-Hellman public key from peer: E - EC Diffie-Hellman public key from peer: E - Module Generat |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | ed Diffie-Hellman public key: G,E - Module Generated Diffie-Hellman private key: G,E - TLS derived key: G,E |
| Zeroization | Zeroize SSP in volatile memory | N/A | SSP | N/A | None | Crypto Officer - AES key: Z - HMAC key: Z - RSA public key: Z - RSA private key: Z - ECDSA public key: Z - ECDSA private key: Z - Diffie-Hellman public key from peer: Z - Diffie-Hellman private key: Z - Diffie-Hellman Shared secret: Z - EC Diffie-Hellman public key |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | from peer: Z - EC Diffie-Hellman private key: Z - EC Diffie-Hellman Shared secret: Z - PBKDF password or passphrase: Z - PBKDF derived key: Z - Entropy input: Z - DRBG seed: Z - DRBG internal state: Z - TLS pre-master secret: Z - TLS master secret: Z - TLS derived key: Z - Intermediate key generation value: Z - HKDF derived key: Z |
| On-Demand Self-test | Initiate power-on self-tests by reset | N/A | N/A | Pass or fail | Symmetric encryption Symmetri | Crypto Officer |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | c decryption Message authentication code (MAC) Message digest Deterministic random bit generation Asymmetric key generation Public key verification Digital signature generation Digital signature verification (Diffie-Hellman) shared secret computation Key derivation (EC Diffie-Hellman) shared secret computation KAS1 (EC Diffie-Hellman) KAS2 (Diffie- | |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | Hellman)<br>Authenticated encryption<br>Authenticated decryption<br>Symmetric Key Generation<br>Digital Signature Verification (Legacy) | |
| On-Demand Integrity Test | Initiate integrity test on-demand | N/A | N/A | Pass or fail | Message authentication code (MAC) | Crypto Officer |
| Show Status | Show status of the module state | N/A | N/A | Module status | None | Crypto Officer |
| Show Module Name and Version | Show the version of the module using gnutls_get_library_config | N/A | N/A | Module name and version | None | Crypto Officer |

Table 14: Approved Services

The “Indicator” column shows the service indicator API functions that must be used to verify the service indicator for each of the services. The function gnutls_fips140_get_operation_state() indicates GNUTLS_FIPS140_OP_APPROVED or GNUTLS_FIPS140_OP_NOT_APPROVED depending on whether the API invoked corresponds to an approved or non-approved algorithm.

## 4.4 Non-Approved Services

| Name | Description | Algorithms | Role |
|---|---|---|---|
| Symmetric encryption;<br>Symmetric decryption | Symmetric encryption;<br>Symmetric decryption | Camellia<br>ChaCha20<br>DES<br>GOST<br>RC2, RC4<br>Salsa20<br>Triple-DES | CO |
| Authenticated encryption; | Authenticated encryption; | Chacha20 and Poly1305<br>AES-GCM (when not used in the context of the TLS protocol) | CO |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Algorithms | Role |
|---|---|---|---|
| Authenticated decryption | Authenticated decryption | | |
| Message authentication code | Message authentication code | CMAC with Triple-DES<br>GMAC<br>HMAC (with keys smaller than 112-bits)<br>HMAC (with GOST)<br>UMAC | CO |
| Message digest | Message digest | GOST<br>MD2, MD5<br>RMD160<br>STREEBOG | CO |
| Key derivation | Key derivation | PBKDF (with non-approved message digest algorithms or using input parameters not meeting requirements stated in section 2.7 of the security policy) | CO |
| Domain parameter generation | Domain parameter generation | DSA | CO |
| Asymmetric key generation | Asymmetric key generation | ECDSA (with curves other than P-256, P-384, P-512)<br>DSA | CO |
| Public key verification | Public key verification | ECDSA (with curves other than P-256, P-384, P-512) | CO |
| Digital signature verification | Digital signature verification | ECDSA (with curves other than P-256, P-384, P-512 or hash functions other than SHA2-224, SHA2-256, SHA2-384, SHA2-512)<br>RSA (with keys smaller than 1024 bits and/or hash functions other than SHA1, SHA2-224, SHA2-256, SHA2-384, SHA2-512)<br>DSA | CO |
| Digital signature generation | Digital signature generation | ECDSA (with curves other than P-256, P-384, P-512 or hash functions other than SHA2-224, SHA2-256, SHA2-384, SHA2-512)<br>RSA (with keys smaller than 2048 bits and/or hash functions other than SHA2-224, SHA2-256, SHA2-384, SHA2-512)<br>DSA | CO |
| Key agreement; Shared secret computation | Key agreement; Shared secret computation | EC Diffie-Hellman (with curves other than P-256, P-384, P-512)<br>SRP<br>Diffie-Hellman (with domain parameters other than safe primes) | CO |
| Key encapsulation; Key un-encapsulation | Key encapsulation; Key un-encapsulation | RSA (with any key sizes) | CO |
| Random number generation | Yarrow | Yarrow | CO |
| Symmetric key generation | Symmetric key generation | Symmetric Keygen (with keys smaller than 112 bits) | CO |

Table 15: Non-Approved Services

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

## 4.5 External Software/Firmware Loaded

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

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 5 Software/Firmware Security

## 5.1 Integrity Techniques

The integrity of the module is verified by comparing an HMAC-SHA2-256 value calculated at run time with the HMAC value stored in the .hmac file that was computed at build time for each software component of the module listed in section 2. If the HMAC values do not match, the test fails, and the module enters the error state.

## 5.2 Initiate on Demand

The pre-operational integrity self-test can be initiated on demand by calling the Self-Test service (via the gnutls_fips140_run_self_tests() function) or by powering-off and reloading the module. During the execution of the on-demand integrity self-test, services are not available, and no data output is possible.

## 5.3 Open-Source Parameters

N/A

## 5.4 Additional Information

N/A

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 6 Operational Environment

## 6.1 Operational Environment Type and Requirements

**Type of Operational Environment**: Modifiable

**How Requirements are Satisfied**:

N/A

## 6.2 Configuration Settings and Restrictions

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

## 6.3 Additional Information

N/A

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 7 Physical Security

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

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 8 Non-Invasive Security

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

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 9 Sensitive Security Parameters Management

## 9.1 Storage Areas

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

Table 16: Storage Areas

## 9.2 SSP Input-Output Methods

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

Table 17: SSP Input-Output Methods

## 9.3 SSP Zeroization Methods

| Zeroization Method | Description | Rationale | Operator Initiation |
|---|---|---|---|
| Wipe and Free memory block allocated | Zeroizes the SSPs contained within the cipher handle. | Memory occupied by SSPs is overwritten with zeroes and then it is released, which renders the SSP values irretrievable. The completion of the zeroization routine indicates that the zeroization has been completed | To zeroize AES keys, call gnutls_cipher_deinit() or gnutls_aead_cipher_deinit(); to zeroize HMAC keys, call gnutls_hmac_deinit(); to zeroize RSA or ECDSA public/private keys, call gnutls_privkey_deinit() or gnutls_x509_privkey_deinit() or gnutls_rsa_params_deinit(); to zeroize Diffie-Hellman public/private keys, call gnutls_dh_params_deinit() or gnutls_pk_params_clear(); to zeroize EC Diffie-Hellman public/private keys, call gnutls_pk_params_clear(); to zeroize (Diffie-Hellman) Shared secret or (EC Diffie-Hellman) Shared secret, call zeroize_key(); to zeroize entropy input, DRBG seed, or DRBG internal state, call gnutls_global_deinit(); to zeroize TLS pre-master secret, TLS master secret, TLS derived key, or PBKDF derived key, call gnutls_deinit() |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Zeroization Method | Description | Rationale | Operator Initiation |
|---|---|---|---|
| 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. The completion of module power-off indicates that the zeroization has been completed | By unloading and reloading the module |

Table 18: SSP Zeroization Methods

All data output is inhibited when the module is performing zeroization.

## 9.4 SSPs

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| AES key | Used for Symmetric encryption; Symmetric decryption; Message authentication code (MAC); | 128, 192, 256 bits - 128, 192, 256 bits | Symmetric key - CSP | | | Symmetric encryption Symmetric decryption Message authentication code (MAC) Authenticated encryption Authenticated decryption |
| Module Generated AES key | Used for Symmetric encryption; Symmetric decryption; Message authentication code (MAC); | 128, 192, 256 bits - 128, 192, 256 bits | Symmetric key - CSP | Symmetric Key Generation | | |
| HMAC key | Used for Message Authentication Code (MAC) | 112 to 524288 bits - 112 to 256 bits | Symmetric key - CSP | | | Message authentication code (MAC) |
| Module Generated HMAC key | Used for Message Authenticati | 112 to 524288 | Symmetric key - CSP | Symmetric Key Generation | | |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | on Code (MAC) | bits - 112 to 256 bits | | | | |
| RSA public key | Used for Digital signature verification; Transport Layer Security (TLS) network protocol | 1024-16384-bits with 112-256 bits of key strength (see section 2.7 for additional information) - 80 to 256 bits | Public key - PSP | | | Digital signature verification TLS v1.2 KDF RFC7627 (A5679) TLS v1.2 KDF RFC7627 (A5939) |
| Module Generated RSA public key | Used for Digital signature verification; Transport Layer Security (TLS) network protocol | 2048, 3072, 4096, 6144, 7680, 8192, 15360-bit modulus (see section 2.7 for additional information) - 112, 128, 149, 178, 196, 201, 256 bits | Public key - PSP | Asymmetric key generation | | Digital signature verification TLS v1.2 KDF RFC7627 (A5679) TLS v1.2 KDF RFC7627 (A5939) |
| RSA private key | Used for Digital signature generation; Transport Layer Security (TLS) network protocol | 2048-16384-bits with 112-256 bits of key strength (see section 2.7 for additional information) - 112 to 256 bits | Private key - CSP | | | Digital signature generation |
| Module Generated RSA private key | Used for Digital signature generation; Transport Layer Security (TLS) | 2048, 3072, 4096, 6144, 7680, 8192, 15360-bit modulus | Private key - CSP | Asymmetric key generation | | Digital signature generation TLS v1.2 KDF RFC7627 (A5679) TLS v1.2 KDF |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | network protocol | (see section 2.7 for additional information) - 112, 128, 149, 178, 196, 201, 256 bits | | | | RFC7627 (A5939) |
| ECDSA public key | Used for Digital signature verification; Public key verification; Transport Layer Security (TLS) network protocol | P-256, P-384, P-521 - 128, 192, 256 bits | Public key - PSP | | | Digital signature verification |
| Module Generated ECDSA public key | Used for Digital signature verification; Public key verification; Transport Layer Security (TLS) network protocol | P-256, P-384, P-521 - 128, 192, 256 bits | Public key - PSP | Asymmetric key generation | | Digital signature verification |
| ECDSA private key | Used for Digital signature generation; Transport Layer Security (TLS) network protocol | P-256, P-384, P-521 - 128, 192, 256 bits | Private key - CSP | | | Digital signature generation |
| Module Generated ECDSA private key | Used for Digital signature generation; Transport Layer Security (TLS) network protocol | P-256, P-384, P-521 - 128, 192, 256 bits | Private key - CSP | Asymmetric key generation | | Digital signature generation |
| Diffie-Hellman public key from peer | Used for Shared secret computation; | ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, | Public key - PSP | | | KAS2 (Diffie-Hellman) |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | Transport Layer Security (TLS) network protocol | ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 - 112 to 200 bits | | | | |
| Module Generated Diffie-Hellman public key | Used for Shared secret computation; Transport Layer Security (TLS) network protocol | ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 - 112 to 200 bits | Public key - PSP | Asymmetric key generation | | KAS2 (Diffie-Hellman) |
| Diffie-Hellman private key | Used for Shared secret computation; Transport Layer Security (TLS) network protocol | ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 - 112 to 200 bits | Private key - CSP | | | (Diffie-Hellman) shared secret computation |
| Module Generated Diffie-Hellman private key | Used for Shared secret computation; Transport Layer Security (TLS) network protocol | ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP- | Private key - CSP | Asymmetric key generation | | KAS2 (Diffie-Hellman) |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | | 4096, MODP-6144, MODP-8192 - 112 to 200 bits | | | | |
| EC Diffie-Hellman public key from peer | Used for Shared secret computation; Transport Layer Security (TLS) network protocol | P-256, P-384, P-521 - 128, 192, 256 bits | Public key - PSP | | | KAS1 (EC Diffie-Hellman) |
| Module Generated EC Diffie-Hellman public key | Used for Shared secret computation; Transport Layer Security (TLS) network protocol | P-256, P-384, P-521 - 128, 192, 256 bits | Public key - PSP | Asymmetric key generation | | KAS1 (EC Diffie-Hellman) |
| EC Diffie-Hellman private key | Used for Shared secret computation; Transport Layer Security (TLS) network protocol | P-256, P-384, P-521 - 128, 192, 256 bits | Private key - CSP | | | (EC Diffie-Hellman) shared secret computation |
| Module Generated EC Diffie-Hellman private key | Used for Shared secret computation; Transport Layer Security (TLS) network protocol | P-256, P-384, P-521 - 128, 192, 256 bits | Private key - CSP | Asymmetric key generation | | KAS1 (EC Diffie-Hellman) |
| Diffie-Hellman Shared secret | Used for Key derivation | ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP- | Shared secret - CSP | | (Diffie-Hellman) shared secret computation KAS2 (Diffie-Hellman) | Key derivation KAS2 (Diffie-Hellman) |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | | 6144, MODP-8192 - 112 to 200 bits | | | | |
| EC Diffie-Hellman Shared secret | Used for Key derivation | P-256, P-384, P-521 - 128, 192, 256 bits | Shared secret - CSP | | (EC Diffie-Hellman) shared secret computation<br>KAS1 (EC Diffie-Hellman) | Key derivation<br>KAS1 (EC Diffie-Hellman) |
| PBKDF password or passphrase | Used for Key derivation | 20 characters or more - N/A | Password - CSP | | | Key derivation |
| PBKDF derived key | Used for protection of storage data | 112 to 256 bits - 112 to 256 bits | Symmetric key - CSP | Key derivation | | |
| Entropy input | Used for Random number generation | 256 to 384 bits - 256 bits | Entropy Input - CSP | | | Deterministic random bit generation |
| DRBG seed | Used for Random number generation | 384 bits - 384 bits | Seed - CSP | Deterministic random bit generation | | Deterministic random bit generation |
| DRBG internal state | Used for Random number generation. This SSP is a CSP in compliance with IG D.L | V: 128 bits; Key: 256 bits - 256 bits | Internal state - CSP | Deterministic random bit generation | | Deterministic random bit generation |
| TLS pre-master secret | Used for Transport Layer Security (TLS) network protocol | ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192, P-256, P-384, P-521 - | Shared secret - CSP | | KAS1 (EC Diffie-Hellman)<br>KAS2 (Diffie-Hellman) | Key derivation |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| | | 112 to 256 bits | | | | |
| TLS master secret | Used for Transport Layer Security (TLS) network protocol | 384 bits - 112 to 256 bits | Intermediate secret value - CSP | Key derivation | KAS1 (EC Diffie-Hellman) KAS2 (Diffie-Hellman) | Key derivation |
| TLS derived key | Used in Transport Layer Security (TLS) network protocol | 112 to 256 bits - 112 to 256 bits | Derived key - CSP | Key derivation | | Symmetric encryption Symmetric decryption Message authentication code (MAC) Authenticated encryption Authenticated decryption |
| Intermediate key generation value | Used in Asymmetric key generation | 256 to 15360 bits - 112 to 256 bits | Intermediate key generation value - CSP | Asymmetric key generation | | Asymmetric key generation |
| HKDF derived key | Key derived from HKDF | 112 to 256 bits - 112 to 256 bits | Symmetric key - CSP | Key derivation | | |

Table 19: SSP Table 1

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| AES key | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | |
| Module Generated AES key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | DRBG internal state:Generated from |
| HMAC key | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | |
| Module Generated HMAC key | API output parameters | RAM:Plaintext | Until explicitly zeroized | Wipe and Free memory block | DRBG internal state:Generated from |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | by operator | allocated Module Reset | |
| RSA public key | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | RSA private key:Paired With |
| Module Generated RSA public key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Module Generated RSA private key:Paired With Intermediate key generation value:Generated from |
| RSA private key | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | RSA public key:Paired With |
| Module Generated RSA private key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Module Generated RSA public key:Paired With Intermediate key generation value:Generated From |
| ECDSA public key | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated | ECDSA private key:Paired With |
| Module Generated ECDSA public key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Module Generated ECDSA private key:Paired With Intermediate key generation value:Generated from |
| ECDSA private key | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | ECDSA public key:Paired With |
| Module Generated ECDSA private key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Module Generated ECDSA public key:Paired With Intermediate key generation value:Generated from |
| Diffie-Hellman public key from peer | API input parameters | RAM:Plaintext | Until explicitly zeroized | Wipe and Free memory | Diffie-Hellman Shared secret:Used to compute |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | by operator | block allocated | |
| Module Generated Diffie-Hellman public key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Diffie-Hellman Shared secret:Used to compute Module Generated Diffie-Hellman private key:Paired With Intermediate key generation value:Generated from |
| Diffie-Hellman private key | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Diffie-Hellman Shared secret:Used to compute Diffie-Hellman public key from peer:Paired With |
| Module Generated Diffie-Hellman private key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Diffie-Hellman Shared secret:Used to compute Module Generated Diffie-Hellman public key:Paired With Intermediate key generation value:Generated from |
| EC Diffie-Hellman public key from peer | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated | EC Diffie-Hellman Shared secret:Used to compute |
| Module Generated EC Diffie-Hellman public key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | EC Diffie-Hellman Shared secret:Used to compute Module Generated EC Diffie-Hellman private key:Paired With Intermediate key generation value:Generated from |
| EC Diffie-Hellman private key | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | EC Diffie-Hellman Shared secret:Used to compute EC Diffie-Hellman public key:Paired With |
| Module Generated EC Diffie-Hellman private key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Diffie-Hellman Shared secret:Used to compute Module Generated EC Diffie-Hellman public key:Paired With Intermediate key |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | | | generation value:Generated from |
| Diffie-Hellman Shared secret | API input parameters API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Diffie-Hellman public key:Computed using Diffie-Hellman private key:Computed using HKDF derived key:Used to derive |
| EC Diffie-Hellman Shared secret | API input parameters API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | EC Diffie-Hellman public key:Computed using EC Diffie-Hellman private key:Computed using HKDF derived key:Used to derive |
| PBKDF password or passphrase | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | PBKDF derived key:Used to derive |
| PBKDF derived key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | PBKDF password or passphrase:Derived From |
| Entropy input | | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | DRBG seed:Used to compute |
| DRBG seed | | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Entropy input:Computed from DRBG internal state:Used to compute |
| DRBG internal state | | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | DRBG seed:Computed from |
| TLS pre-master secret | API input parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | TLS master secret:Used to compute |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| TLS master secret | | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | TLS pre-master secret:Computed from TLS derived key:Used to derive |
| TLS derived key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | TLS master secret:Derived From |
| Intermediate key generation value | | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | RSA public key:Intermediate value obtained during generated of RSA private key:Intermediate value obtained during generated of ECDSA public key:Intermediate value obtained during generated of ECDSA private key:Intermediate value obtained during generated of Diffie-Hellman public key:Intermediate value obtained during generated of Diffie-Hellman private key:Intermediate value obtained during generated of EC Diffie-Hellman public key:Intermediate value obtained during generated of EC Diffie-Hellman private key:Intermediate value obtained during generated of |
| HKDF derived key | API output parameters | RAM:Plaintext | Until explicitly zeroized by operator | Wipe and Free memory block allocated Module Reset | Diffie-Hellman Shared secret:Derived From EC Diffie-Hellman Shared secret:Derived From |

Table 20: SSP Table 2

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

## 9.5 Transitions

- SHA-1 will be disallowed for all uses besides digital signature verification on January 1st 2031.

## 9.6 Additional Information

N/A

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 10 Self-Tests

The module performs the pre-operational self-test and cryptographic algorithm self-tests (CASTs) automatically when the module is loaded into memory. The module's services are not available for use and data input and output are inhibited until the pre-operational tests and CASTs are completed successfully. If any of the pre-operational integrity self-tests or CASTs fail, an error message is returned and the module transitions to the error state.

## 10.1 Pre-Operational Self-Tests

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details |
|---|---|---|---|---|---|
| HMAC-SHA2-256 (A5674) | SHA2-256 | Integrity | SW/FW Integrity | Module becomes operational and services are available for use | MAC tag computation |
| HMAC-SHA2-256 (A5679) | SHA2-256 | Integrity | SW/FW Integrity | Module becomes operational and services are available for use | MAC tag computation |
| HMAC-SHA2-256 (A5713) | SHA2-256 | Integrity | SW/FW Integrity | Module becomes operational and services are available for use | MAC tag computation |

Table 21: Pre-Operational Self-Tests

## 10.2 Conditional Self-Tests

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-CBC (A5671) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A5672) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A5673) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A5674) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A5679) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A5713) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-CFB8 (A5676) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB8 (A5677) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB8 (A5682) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A5671) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A5672) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A5673) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A5674) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A5679) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A5713) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| RSA SigGen (FIPS186-5) (A5679) | 2048-bit key using SHA2-256 | KAT | CAST | Module becomes operational | Digital signature generation | Test runs at power-on before the integrity test |
| RSA SigVer (FIPS186-5) (A5679) | 2048-bit key using SHA2-256 | KAT | CAST | Module becomes operational | Digital signature verification | Test runs at power-on before the integrity test |
| AES-XTS Testing Revision 2.0 (A5680) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| KAS-FFC-SSC Sp800-56Ar3 (A5679) | 3072-bit key and safe prime ffdhe3072 | KAT | CAST | Module becomes operational | Shared secret computation | Test runs at power-on before the integrity test |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| Counter DRBG (A5679) | 256-bit key without DF, without PR | KAT | CAST | Module becomes operational | SP 800-90Ar1 (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| KAS-ECC-SSC Sp800-56Ar3 (A5679) | P-256 | KAT | CAST | Module becomes operational | Shared secret computation | Test runs at power-on before the integrity test |
| ECDSA SigGen (FIPS186-5) (A5679) | P-256 using SHA2-256 | KAT | CAST | Module becomes operational | Digital signature generation | Test runs at power-on before the integrity test |
| ECDSA SigVer (FIPS186-5) (A5679) | P-256 using SHA2-256 | KAT | CAST | Module becomes operational | Digital signature verification | Test runs at power-on before the integrity test |
| KDA HKDF Sp800-56Cr1 (A5678) | SHA2-256 | KAT | CAST | Module becomes operational | Shared secret key derivation | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A5674) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A5679) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A5713) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A5674) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A5679) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A5713) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A5674) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A5679) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| HMAC-SHA2-256 (A5713) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A5674) | SHA-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A5679) | SHA-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A5713) | SHA-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A5674) | SHA-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A5679) | SHA-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A5713) | SHA-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| PBKDF (A5679) | HMAC-SHA2-256 | KAT | CAST | Module becomes operational | Password-based key derivation | Test runs at power-on before the integrity test |
| SHA3-224 (A5675) | SHA3-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-224 (A5681) | SHA3-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-256 (A5675) | SHA3-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-256 (A5681) | SHA3-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-384 (A5675) | SHA3-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| SHA3-384 (A5681) | SHA3-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-512 (A5675) | SHA3-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-512 (A5681) | SHA3-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| TLS v1.2 KDF RFC7627 (A5679) | SHA2-256 | KAT | CAST | Module becomes operational | Industry-based TLS v1.2 KDF key derivation | Test runs at power-on before the integrity test |
| ECDSA KeyGen (FIPS186-5) (A5679) | SHA2-256 | PCT | PCT | Successful key pair generation | Signature generation & verification | Key pair generation |
| RSA KeyGen (FIPS186-5) (A5679) | SHA2-256 | PCT | PCT | Successful key pair generation | Signature generation & verification | Key pair generation |
| Safe Primes Key Generation (A5679) | N/A | PCT | PCT | Successful key pair generation | Section 5.6.2.1.4 of SP800-56Arev3 | Key pair generation |
| AES-CBC (A5935) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A5936) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A5939) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A5935) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB8 (A5942) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A5936) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-GCM (A5939) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-XTS Testing Revision 2.0 (A5940) | 256-bit key, encrypt/decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| Counter DRBG (A5939) | 256-bit key without DF, without PR | KAT | CAST | Module becomes operational | SP 800-90Ar1 (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| ECDSA KeyGen (FIPS186-5) (A5939) | SHA2-256 | PCT | PCT | Successful key pair generation | Signature generation & verification | Key pair generation |
| KAS-ECC-SSC Sp800-56Ar3 (A5939) | P-256 | KAT | CAST | Module becomes operational | Shared secret computation | Test runs at power-on before the integrity test |
| ECDSA SigGen (FIPS186-5) (A5939) | P-256 using SHA2-256 | KAT | CAST | Module becomes operational | Digital signature generation | Test runs at power-on before the integrity test |
| ECDSA SigVer (FIPS186-5) (A5939) | P-256 using SHA2-256 | KAT | CAST | Module becomes operational | Digital signature verification | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A5937) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A5939) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A5939) | SHA2-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A5937) | SHA2-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A5937) | SHA2-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A5939) | SHA2-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| HMAC-SHA2-512 (A5937) | SHA2-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A5939) | SHA2-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| KDA HKDF Sp800-56Cr1 (A5938) | SHA2-256 | KAT | CAST | Module becomes operational | Shared secret key derivation | Test runs at power-on before the integrity test |
| KDF TLS (A5679) | HMAC-SHA2-256 | KAT | CAST | Module becomes operational | Industry-based TLS KDF key derivation | Test runs at power-on before the integrity test |
| KDF TLS (A5939) | HMAC-SHA2-256 | KAT | CAST | Module becomes operational | Industry-based TLS KDF key derivation | Test runs at power-on before the integrity test |
| PBKDF (A5939) | HMAC-SHA2-256 | KAT | CAST | Module becomes operational | Password-based key derivation | Test runs at power-on before the integrity test |
| RSA KeyGen (FIPS186-5) (A5939) | SHA2-256 | PCT | PCT | Successful key pair generation | Signature generation & verification | Key pair generation |
| RSA SigGen (FIPS186-5) (A5939) | 2048-bit key using SHA2-256 | KAT | CAST | Module becomes operational | Digital signature generation | Test runs at power-on before the integrity test |
| RSA SigVer (FIPS186-5) (A5939) | 2048-bit key using SHA2-256 | KAT | CAST | Module becomes operational | Digital signature verification | Test runs at power-on before the integrity test |
| Safe Primes Key Generation (A5939) | N/A | PCT | PCT | Successful key pair generation | Section 5.6.2.1.4 of SP800-56Arev3 | Key pair generation |
| HMAC-SHA2-256 (A5937) | SHA2-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| SHA3-224 (A5941) | SHA3-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-256 (A5941) | SHA3-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| SHA3-384 (A5941) | SHA3-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-512 (A5941) | SHA3-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| TLS v1.2 KDF RFC7627 (A5939) | SHA2-256 | KAT | CAST | Module becomes operational | Industry-based TLS v1.2 KDF key derivation | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A5939) | SHA2-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| KAS-FFC-SSC Sp800-56Ar3 (A5939) | 3072-bit key and safe prime ffdhe3072 | KAT | CAST | Module becomes operational | Shared secret computation | Test runs at power-on before the integrity test |

Table 22: Conditional Self-Tests

## 10.3 Periodic Self-Test Information

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| HMAC-SHA2-256 (A5674) | Integrity | SW/FW Integrity | On demand | Manual |
| HMAC-SHA2-256 (A5679) | Integrity | SW/FW Integrity | On demand | Manual |
| HMAC-SHA2-256 (A5713) | Integrity | SW/FW Integrity | On demand | Manual |

Table 23: Pre-Operational Periodic Information

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-CBC (A5671) | KAT | CAST | On Demand | Manually |
| AES-CBC (A5672) | KAT | CAST | On Demand | Manually |
| AES-CBC (A5673) | KAT | CAST | On Demand | Manually |
| AES-CBC (A5674) | KAT | CAST | On Demand | Manually |
| AES-CBC (A5679) | KAT | CAST | On Demand | Manually |
| AES-CBC (A5713) | KAT | CAST | On Demand | Manually |
| AES-CFB8 (A5676) | KAT | CAST | On Demand | Manually |
| AES-CFB8 (A5677) | KAT | CAST | On Demand | Manually |
| AES-CFB8 (A5682) | KAT | CAST | On Demand | Manually |
| AES-GCM (A5671) | KAT | CAST | On Demand | Manually |
| AES-GCM (A5672) | KAT | CAST | On Demand | Manually |
| AES-GCM (A5673) | KAT | CAST | On Demand | Manually |
| AES-GCM (A5674) | KAT | CAST | On Demand | Manually |
| AES-GCM (A5679) | KAT | CAST | On Demand | Manually |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-GCM (A5713) | KAT | CAST | On Demand | Manually |
| RSA SigGen (FIPS186-5) (A5679) | KAT | CAST | On Demand | Manually |
| RSA SigVer (FIPS186-5) (A5679) | KAT | CAST | On Demand | Manually |
| AES-XTS Testing Revision 2.0 (A5680) | KAT | CAST | On Demand | Manually |
| KAS-FFC-SSC Sp800-56Ar3 (A5679) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A5679) | KAT | CAST | On Demand | Manually |
| KAS-ECC-SSC Sp800-56Ar3 (A5679) | KAT | CAST | On Demand | Manually |
| ECDSA SigGen (FIPS186-5) (A5679) | KAT | CAST | On Demand | Manually |
| ECDSA SigVer (FIPS186-5) (A5679) | KAT | CAST | On Demand | Manually |
| KDA HKDF Sp800-56Cr1 (A5678) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A5674) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A5679) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A5713) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A5674) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A5679) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A5713) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A5674) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A5679) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A5713) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A5674) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A5679) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A5713) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A5674) | KAT | CAST | On Demand | Manually |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| HMAC-SHA2-512 (A5679) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A5713) | KAT | CAST | On Demand | Manually |
| PBKDF (A5679) | KAT | CAST | On Demand | Manually |
| SHA3-224 (A5675) | KAT | CAST | On Demand | Manually |
| SHA3-224 (A5681) | KAT | CAST | On Demand | Manually |
| SHA3-256 (A5675) | KAT | CAST | On Demand | Manually |
| SHA3-256 (A5681) | KAT | CAST | On Demand | Manually |
| SHA3-384 (A5675) | KAT | CAST | On Demand | Manually |
| SHA3-384 (A5681) | KAT | CAST | On Demand | Manually |
| SHA3-512 (A5675) | KAT | CAST | On Demand | Manually |
| SHA3-512 (A5681) | KAT | CAST | On Demand | Manually |
| TLS v1.2 KDF RFC7627 (A5679) | KAT | CAST | On Demand | Manually |
| ECDSA KeyGen (FIPS186-5) (A5679) | PCT | PCT | On Demand | Manually |
| RSA KeyGen (FIPS186-5) (A5679) | PCT | PCT | On Demand | Manually |
| Safe Primes Key Generation (A5679) | PCT | PCT | On Demand | Manually |
| AES-CBC (A5935) | KAT | CAST | On Demand | Manually |
| AES-CBC (A5936) | KAT | CAST | On Demand | Manually |
| AES-CBC (A5939) | KAT | CAST | On Demand | Manually |
| AES-GCM (A5935) | KAT | CAST | On Demand | Manually |
| AES-CFB8 (A5942) | KAT | CAST | On Demand | Manually |
| AES-GCM (A5936) | KAT | CAST | On Demand | Manually |
| AES-GCM (A5939) | KAT | CAST | On Demand | Manually |
| AES-XTS Testing Revision 2.0 (A5940) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A5939) | KAT | CAST | On Demand | Manually |
| ECDSA KeyGen (FIPS186-5) (A5939) | PCT | PCT | On Demand | Manually |
| KAS-ECC-SSC Sp800-56Ar3 (A5939) | KAT | CAST | On Demand | Manually |
| ECDSA SigGen (FIPS186-5) (A5939) | KAT | CAST | On Demand | Manually |
| ECDSA SigVer (FIPS186-5) (A5939) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A5937) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A5939) | KAT | CAST | On Demand | Manually |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| HMAC-SHA2-224 (A5939) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A5937) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A5937) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A5939) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A5937) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A5939) | KAT | CAST | On Demand | Manually |
| KDA HKDF Sp800-56Cr1 (A5938) | KAT | CAST | On Demand | Manually |
| KDF TLS (A5679) | KAT | CAST | On Demand | Manually |
| KDF TLS (A5939) | KAT | CAST | On Demand | Manually |
| PBKDF (A5939) | KAT | CAST | On Demand | Manually |
| RSA KeyGen (FIPS186-5) (A5939) | PCT | PCT | On Demand | Manually |
| RSA SigGen (FIPS186-5) (A5939) | KAT | CAST | On Demand | Manually |
| RSA SigVer (FIPS186-5) (A5939) | KAT | CAST | On Demand | Manually |
| Safe Primes Key Generation (A5939) | PCT | PCT | On Demand | Manually |
| HMAC-SHA2-256 (A5937) | KAT | CAST | On Demand | Manually |
| SHA3-224 (A5941) | KAT | CAST | On Demand | Manually |
| SHA3-256 (A5941) | KAT | CAST | On Demand | Manually |
| SHA3-384 (A5941) | KAT | CAST | On Demand | Manually |
| SHA3-512 (A5941) | KAT | CAST | On Demand | Manually |
| TLS v1.2 KDF RFC7627 (A5939) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A5939) | KAT | CAST | On Demand | Manually |
| KAS-FFC-SSC Sp800-56Ar3 (A5939) | KAT | CAST | On Demand | Manually |

Table 24: Conditional Periodic Information

## 10.4 Error States

| Name | Description | Conditions | Recovery Method | Indicator |
|---|---|---|---|---|
| Error State | Prevents any cryptographic related | When the integrity test or KAT (not the | Restarting the module | GNUTLS_E_SELF_TEST_ERROR; GNUTLS_E_RANDOM_FAILED; |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Name | Description | Conditions | Recovery Method | Indicator |
|---|---|---|---|---|
| | operations and data output | DRBG KAT) fail When the DRBG KAT fails When a newly generated RSA, ECDSA, Diffie- Hellman or EC Diffie-Hellman key pair fails the PCT | | GNUTLS_E_PK_GENERATION_ERROR; GNUTLS_E_LIB_IN_ERROR_STATE |

Table 25: Error States

The calling application can obtain the module state by calling the gnutls_fips140_get_operation_state() API function. The function returns GNUTLS_FIPS140_OP_ERROR if the module is in the Error state.

## 10.5 Operator Initiation of Self-Tests

The operator can initiate the pre-operational integrity self-test and cryptographic algorithm self-tests by calling the Self-Test service (via the gnutls_fips140_run_self_tests() function) or by powering-off and reloading the module. The operator can initiate a pairwise consistency self-test by calling the “Asymmetric key generation” service. During the execution of the pre-operational integrity self-test and cryptographic algorithm self-tests, services are not available, and no data output is possible.

## 10.6 Additional Information

N/A

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 11 Life-Cycle Assurance

## 11.1 Installation, Initialization, and Startup Procedures

11.1.1 Configuration of the Operating Environment

The module needs to be set to run in the FIPS validated operational environment. This can be enabled automatically via the Ubuntu Advantage tool after attaching your subscription.

(1) To install the tool, type the following commands:

$ sudo apt update

$ sudo apt install ubuntu-advantage-tools

(2) To activate the Ubuntu Pro subscription run:

$ sudo pro attach <your_pro_token>

(3) To enable the FIPS validated operational environment run:

$ sudo pro enable fips

(4) To verify that the FIPS validated operational environment is enabled run:

$ sudo pro status

The pro client will install the necessary packages that are part of the FIPS validated operational environment, including the kernel and the bootloader. After this step you MUST reboot to enter the FIPS validated operational environment. The reboot will boot into the kernel of the FIPS validated operational environment and create the /proc/sys/crypto/fips_enabled entry which tells the FIPS certified modules to run in the approved mode of operation. If you do not reboot after installing and configuring the bootloader, you will not be in the FIPS validated operational environment.

To verify that the FIPS validated operational environment is enabled after the reboot check the /proc/sys/crypto/fips_enabled file and ensure it is set to 1. If it is set to 0, the FIPS modules will not run in the approved mode of operation. If the file is missing, the correct kernel (which is part of the FIPS validated operational environment) is not installed. You can verify that the FIPS validated operational environment has been properly enabled with the pro status command.

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

If the module is not installed, initialized, and configured according to this section, the module is in a non-compliant state. If the module is in a non-compliant state, it can be placed into the compliant state by un-initializing and uninstalling the module and then installing, initializing, and configuring the module according to this section.

### 11.1.2 Delivery of the Module

On the Supermicro SYS-1019P-WTR hardware platform with the Intel Xeon Gold 6226 processor, the module is delivered through the following Ubuntu packages:

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

libgnutls-dane0t64_3.8.3-1.1ubuntu3.1+Fips1_amd64.deb
libnettle8t64_3.9.1-2.2build1.1_amd64.deb
libhogweed6t64_3.9.1-2.2build1.1_amd64.deb
libgmp10_6.3.0+dfsg-2ubuntu6_amd64.deb

On the Amazon Web Services (AWS) c6g.metal hardware platform with the AWS Graviton3 processor, the module is delivered through the following Ubuntu packages:

libgnutls-dane0t64_3.8.3-1.1ubuntu3.1+Fips1_arm64.deb
libnettle8t64_3.9.1-2.2build1.1_ arm64.deb
libhogweed6t64_3.9.1-2.2build1.1_ arm64.deb
libgmp10_6.3.0+dfsg-2ubuntu6_ arm64.deb

On the IBM z16 hardware platform with the IBM Telum processor, the module is delivered through the following Ubuntu packages:

libgnutls-dane0t64_3.8.3-1.1ubuntu3.1+Fips1_ s390x.deb
libnettle8t64_3.9.1-2.2build1.1_ s390x.deb
libhogweed6t64_3.9.1-2.2build1.1_ s390x.deb
libgmp10_6.3.0+dfsg-2ubuntu6_ s390x.deb

### 11.1.3 Installation of the Module

After the operating environment has been configured according to the instructions of section 11.1.1, the Crypto Officer can install the Ubuntu packages containing the module using the Advanced Package Tool (APT) with the following commands:

$ sudo apt-get install libgnutls30=3.8.3-1.1ubuntu3.1+Fips1

$ sudo apt-get install libgmp10=2:6.3.0+dfsg-2ubuntu6+Fips1

$ sudo apt-get install libhogweed6=3.9.1-2.2build1.1

$ sudo apt-get install libnettle8t64=3.9.1-2.2build1.1

All the Ubuntu packages are associated with hashes for integrity check. The integrity of the Ubuntu package is automatically verified by the packing tool during the installation of the module. The Crypto Officer shall not install the package if the integrity fails.

The module cannot use the following environment variables:

GNUTLS_NO_EXPLICIT_INIT
GNUTLS_SKIP_FIPS_INTEGRITY_CHECKS

The module can only be used with the cryptographic algorithms provided. Therefore, the following API functions are forbidden in the approved mode of operation:

gnutls_privkey_import_ext4

## 11.2 Administrator Guidance

The Crypto Officer shall follow this Security Policy to configure the operational environment and install the module to be operated in the approved mode.

The output of the “Show module name and version” service is:

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice. 

“Canonical Ltd. Ubuntu 24.04 GnuTLS Cryptographic Module” and "3.8.3-1.1ubuntu3.1+Fips1".

## 11.3 Non-Administrator Guidance

The approved security functions are listed in section 2.6 of this security policy.

The logical interfaces available to the users of the cryptographic module are listed in section 3.1.

For the secure operation of the module, the operator must follow the instructions in section 11.1 of this security policy.

## 11.4 Design and Rules

N/A

## 11.5 Maintenance Requirements

N/A

## 11.6 End of Life

For secure sanitization of the cryptographic module, the module needs first to be powered off, which will zeroize all keys and CSPs in volatile memory. The module does not possess persistent storage of SSPs, so further sanitization steps are not needed.

## 11.7 Additional Information

N/A

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# 12 Mitigation of Other Attacks

The module does not mitigate other attacks.

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# Appendix A: TLS Cipher Suites

The module supports the following cipher suites for the TLS protocol version 1.0, 1.1, 1.2 and 1.3, compliant with section 3.3.1 of SP 800-52 Rev. 2. Each cipher suite defines the key exchange algorithm, the bulk encryption algorithm (including the symmetric key size) and the MAC algorithm.

| Cipher Suite | ID | Reference |
|---|---|---|
| TLS_DH_RSA_WITH_AES_128_CBC_SHA | { 0x00, 0x31 } | RFC3268 |
| TLS_DHE_RSA_WITH_AES_128_CBC_SHA | { 0x00, 0x33 } | RFC3268 |
| TLS_DH_RSA_WITH_AES_256_CBC_SHA | { 0x00, 0x37 } | RFC3268 |
| TLS_DHE_RSA_WITH_AES_256_CBC_SHA | { 0x00, 0x39 } | RFC3268 |
| TLS_DH_RSA_WITH_AES_128_CBC_SHA256 | { 0x00,0x3F } | RFC5246 |
| TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 | { 0x00,0x67 } | RFC5246 |
| TLS_DH_RSA_WITH_AES_256_CBC_SHA256 | { 0x00,0x69 } | RFC5246 |
| TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 | { 0x00,0x6B } | RFC5246 |
| TLS_PSK_WITH_AES_128_CBC_SHA | { 0x00, 0x8C } | RFC4279 |
| TLS_PSK_WITH_AES_256_CBC_SHA | { 0x00, 0x8D } | RFC4279 |
| TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 | { 0x00, 0x9E } | RFC5288 |
| TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 | { 0x00, 0x9F } | RFC5288 |
| TLS_DH_RSA_WITH_AES_128_GCM_SHA256 | { 0x00, 0xA0 } | RFC5288 |
| TLS_DH_RSA_WITH_AES_256_GCM_SHA384 | { 0x00, 0xA1 } | RFC5288 |
| TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA | { 0xC0, 0x04 } | RFC4492 |
| TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA | { 0xC0, 0x05 } | RFC4492 |
| TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA | { 0xC0, 0x09 } | RFC4492 |
| TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA | { 0xC0, 0x0A } | RFC4492 |
| TLS_ECDH_RSA_WITH_AES_128_CBC_SHA | { 0xC0, 0x0E } | RFC4492 |
| TLS_ECDH_RSA_WITH_AES_256_CBC_SHA | { 0xC0, 0x0F } | RFC4492 |
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA | { 0xC0, 0x13 } | RFC4492 |
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA | { 0xC0, 0x14 } | RFC4492 |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| Cipher Suite | ID | Reference |
|---|---|---|
| TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 | { 0xC0, 0x23 } | RFC5289 |
| TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 | { 0xC0, 0x24 } | RFC5289 |
| TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 | { 0xC0, 0x25 } | RFC5289 |
| TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 | { 0xC0, 0x26 } | RFC5289 |
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 | { 0xC0, 0x27 } | RFC5289 |
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 | { 0xC0, 0x28 } | RFC5289 |
| TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256 | { 0xC0, 0x29 } | RFC5289 |
| TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384 | { 0xC0, 0x2A } | RFC5289 |
| TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 | { 0xC0, 0x2B } | RFC5289 |
| TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 | { 0xC0, 0x2C } | RFC5289 |
| TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 | { 0xC0, 0x2D } | RFC5289 |
| TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 | { 0xC0, 0x2E } | RFC5289 |
| TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 | { 0xC0, 0x2F } | RFC5289 |
| TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 | { 0xC0, 0x30 } | RFC5289 |
| TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 | { 0xC0, 0x31 } | RFC5289 |
| TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384 | { 0xC0, 0x32 } | RFC5289 |
| TLS_DHE_RSA_WITH_AES_128_CCM | { 0xC0, 0x9E } | RFC6655 |
| TLS_DHE_RSA_WITH_AES_256_CCM | { 0xC0, 0x9F } | RFC6655 |
| TLS_DHE_RSA_WITH_AES_128_CCM_8 | { 0xC0, 0xA2 } | RFC6655 |
| TLS_DHE_RSA_WITH_AES_256_CCM_8 | { 0xC0, 0xA3 } | RFC6655 |
| TLS_AES_128_GCM_SHA256 | { 0x13, 0x01 } | RFC8446 |
| TLS_AES_256_GCM_SHA384 | { 0x13, 0x02 } | RFC8446 |
| TLS_AES_128_CCM_SHA256 | { 0x13, 0x04 } | RFC8446 |
| TLS_AES_128_CCM_8_SHA256 | { 0x13, 0x05 } | RFC8446 |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# Appendix B. Glossary and Abbreviations

| | |
|---|---|
| **AES** | Advanced Encryption Standard |
| **AES-NI** | Advanced Encryption Standard New Instructions |
| **BC-AUTH** | Authenticated Block Cipher |
| **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 |
| **CO** | Crypto Officer |
| **CPACF** | Central Processor Assist for Cryptographic Function |
| **CSP** | Critical Security Parameter |
| **CTR** | Counter Mode |
| **CVL** | Component Validation List |
| **DF** | Derivation Function |
| **DPK** | Data Protection Key |
| **DRBG** | Deterministic Random Bit Generator |
| **ECB** | Electronic Code Book |
| **ECC** | Elliptic Curve Cryptography |
| **FFC** | Finite Field Cryptography |
| **FIPS** | Federal Information Processing Standards Publication |
| **FSM** | Finite State Model |
| **GCM** | Galois Counter Mode |
| **HMAC** | Hash Message Authentication Code |
| **KAS** | Key Agreement Schema |
| **KAS-SSC** | Key Agreement Schema Secure Secret Computation |
| **KAT** | Known Answer Test |
| **KDF** | Key Derivation Function |
| **MAC** | Message Authentication Code |
| **MK** | Master Key |
| **NIST** | National Institute of Science and Technology |
| **OFB** | Output Feedback |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| | |
|---|---|
| **OS** | Operating System |
| **PAA** | Processor Algorithm Acceleration |
| **PBKDF** | Password Based Key Derivation Function |
| **PAI** | Processor Algorithm Implementation |
| **PCT** | Pairwise-Consistency Test |
| **PR** | Prediction Resistance |
| **PSS** | Probabilistic Signature Scheme |
| **PSP** | Public Security Parameter |
| **RNG** | Random Number Generator |
| **RSA** | Rivest, Shamir, Addleman |
| **SHA** | Secure Hash Algorithm |
| **SHS** | Secure Hash Standard |
| **SSH** | Secure Shell |
| **SSP** | Sensitive Security Parameter |
| **TLS** | Transport Layer Security |
| **XTS** | XEX-based Tweaked-codebook mode with cipher text Stealing |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

# Appendix C. References


| | |
|---|---|
| **FIPS 140-3** | **FIPS PUB 140-3 - Security Requirements For Cryptographic Modules**<br>March 2019<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf |
| **FIPS 140-3 IG** | **Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program**<br>October 2024<br>https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf |
| **FIPS 180-4** | **Secure Hash Standard (SHS)**<br>August 2015<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf |
| **FIPS 186-4** | **Digital Signature Standard (DSS)**<br>July 2013<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf |
| **FIPS 186-5** | **Digital Signature Standard (DSS)**<br>February 2023<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf |
| **FIPS 197** | **Advanced Encryption Standard**<br>November 2001<br>https://csrc.nist.gov/publications/fips/fips197/fips-197.pdf |
| **FIPS 198-1** | **The Keyed Hash Message Authentication Code (HMAC)**<br>July 2008<br>https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf |
| **FIPS 202** | **SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions**<br>August 2015<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf |
| **PKCS#1** | **Public Key Cryptography Standards (PKCS) #1: RSA Cryptography**<br>Specifications Version 2.1<br>February 2003<br>https://www.ietf.org/rfc/rfc3447.txt |
| **RFC3394** | **Advanced Encryption Standard (AES) Key Wrap Algorithm**<br>September 2002<br>https://www.ietf.org/rfc/rfc3394.txt |
| **RFC5649** | **Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm**<br>September 2009<br>https://www.ietf.org/rfc/rfc5649.txt |


© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| | |
|---|---|
| **SP 800-38A** | **NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques**<br>December 2001<br>https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf |
| **SP 800-38B** | **NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication**<br>May 2005<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38b.pdf |
| **SP 800-38C** | **NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality**<br>May 2004<br>https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf |
| **SP 800-38D** | **NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC**<br>November 2007<br>https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf |
| **SP 800-38E** | **NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices**<br>January 2010<br>https://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf |
| **SP 800-38F** | **NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping**<br>December 2012<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf |
| **SP 800-38G** | **NIST Special Publication 800-38G - Recommendation for Block Cipher Modes of Operation: Methods for Format - Preserving Encryption**<br>March 2016<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf |
| **SP 800-52 Rev. 2** | **NIST Special Publication 800-52 Revision 2 - Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations**<br>August 2019<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf |
| **SP 800-56A Rev. 3** | **NIST Special Publication 800-56A Revision 3 - Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptography**<br>April 2018<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf |
| **SP 800-56C Rev. 2** | **NIST Special Publication 800-56C Revision 2 - Recommendation for Key Derivation through Extraction-then-Expansion**<br>August 2020<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

| | |
|---|---|
| **SP 800-57 Rev. 5** | **NIST Special Publication 800-57 Part 1 Revision 5 - Recommendation for Key Management Part 1: General**<br>May 2020<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf |
| **SP 800-90A Rev. 1** | **NIST Special Publication 800-90A Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators**<br>June 2015<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf |
| **SP 800-90B** | **NIST Special Publication 800-90B - Recommendation for the Entropy Sources Used for Random Bit Generation**<br>January 2018<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf |
| **SP 800-108 Rev. 1** | **NIST Special Publication 800-108 Revision 1 - Recommendation for Key Derivation Using Pseudorandom Functions (Revised)**<br>August 2022<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1-upd1.pdf |
| **SP 800-131A Rev. 2** | **NIST Special Publication 800-131 Revision 2 - Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths**<br>March 2019<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf |
| **SP 800-132** | **NIST Special Publication 800-132 - Recommendation for Password-Based Key Derivation - Part 1: Storage Applications**<br>December 2010<br>https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf |
| **SP 800-133 Rev. 2** | **NIST Special Publication 800-133 Revision 2 - Recommendation for Cryptographic Key Generation**<br>June 2020<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf |
| **SP 800-135 Rev. 1** | **NIST Special Publication 800-135 Revision 1 - Recommendation for Existing Application-Specific Key Derivation Functions**<br>December 2011<br>https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf |
| **SP 800-140B Rev. 1** | **NIST Special Publication 800-140B Revision 1 - CMVP Security Policy Requirements**<br>November 2023<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140Br1.pdf |

© 2026 Canonical Ltd. / atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.