UNISYS | Securing Your Tomorrow®

Unisys Linux OpenSSL FIPS Object Module Version 2.0

FIPS 140-2 Level 1 Validation Non-Proprietary Security Policy

May 27, 2021

Table of Contents

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

1. Introduction

1.1. Document History

Authors	*Date*	*Version*	*Comment*
Unisys Stealth Team	June 5, 2019	0.1	Initial draft.
Unisys Stealth Team	June 26, 2019	0.2	Updated draft.
Unisys Stealth Team	June 1, 2020	0.3	Final draft.
Unisys Stealth Team	March 24, 2021	0.4	Updated draft.

1.2. Purpose

This is a non-proprietary FIPS 140-2 Security Policy for the Unisys Linux OpenSSL FIPS Object Module, which is referred to as *the module*. This document describes how this module meets all of the requirements as specified in the Federal Information Processing Standards (FIPS) Publication 140-2. This document also describes how to run the module in a secure, FIPS-approved mode of operation. This Policy forms a part of the submission package to the validating lab.

FIPS 140-2 specifies the security requirements for a cryptographic module protecting sensitive information. Based on four security levels for cryptographic modules this standard identifies requirements in ten sections. For more information about the standard visit www.nist.gov/cmvp.

The product meets the overall requirements applicable to Level 1 security for FIPS 140-2. The module does not support authentication mechanisms.

Table 1 provides a list of the security requirement sections and their associated levels.

Table 1 – Module Compliance Table

Security Requirement	Security Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A
Cryptographic Module Security Policy	1
Overall Level of Certification	1

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

2. Cryptographic Module Description

The module is a software library providing a C-language Application Program Interface (API) for use by other processes that require cryptographic functionality. The module is classified by FIPS 140-2 as a software module, multi-chip standalone module embodiment.

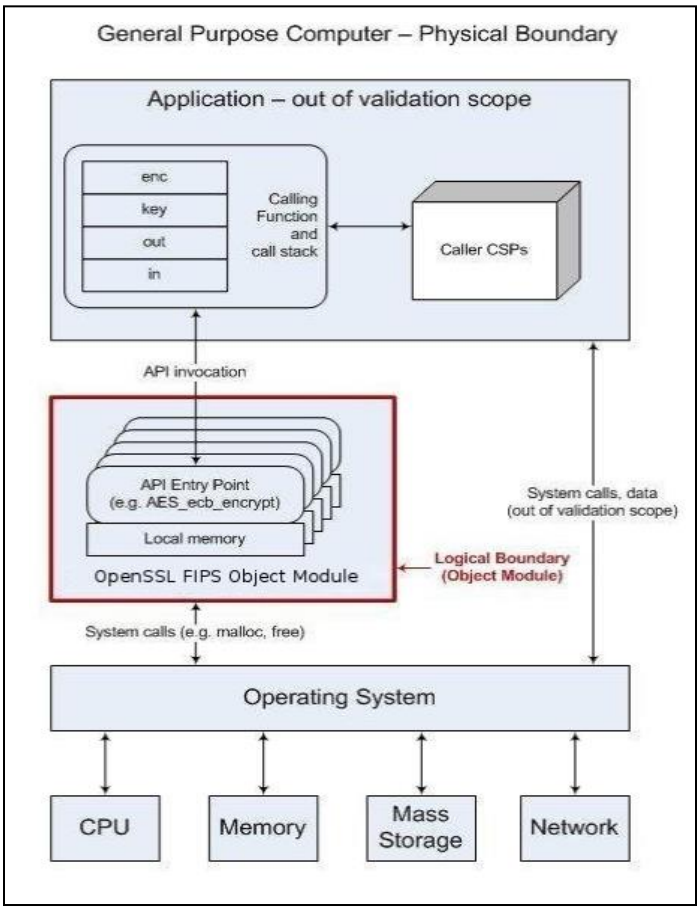
The module implements Triple-DES, AES, SHA, SHA2, DRNG, DSA, RSA, HMAC, CCM, ECDSA (prime curves only), CMAC, KAS ECC (prime curves only) and GCM algorithms in the approved mode.

2.1. Cryptographic Boundary

The physical cryptographic boundary is the general purpose computer on which the module is installed. The logical cryptographic boundary of the module is the fipscanister.o object module, which is embedded in the Unisys openssl libcrypto.so dynamic library. The module performs no communications other than with the calling application (the process that invokes the module services).

Figure 1 is the software block diagram of the module, and it illustrates the module boundary.

Figure 1 – Software Block Diagram



© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

2.2. Tested Configurations

Table 2 describes the multi-chip standalone platforms on which the module has been tested.

Table 2 – Tested Operational Environments

Manufacturer	Model	Operating System
Intel® Xeon® Gold 5115 Processor	Dell PowerEdge R640 Server	Ubuntu 18.04 LTS Server distribution

The module has only been tested on the configurations listed above. The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment not listed above or on the validation certificate (see IG G.5).

3. Module Ports and Interfaces

The module is considered to be a multi-chip standalone module designed to meet FIPS 140-2 Level 1 requirements. The physical ports of the module are the same as the computer system on which the software module is executing. The logical interface is an API as shown in Table 3.

Table 3 – Mapping Physical and Logical Interfaces

FIPS 140-2 Logical Interface	Description
Data Input	API entry point data input stack parameters
Data Output	API entry point data output stack parameters
Control Input	API entry point and corresponding stack parameters
Status Output	API entry point return values and status stack parameters

When the module is performing self-tests or is in an error state, all output on the logical data output interface is inhibited. As a software module, it cannot control the physical ports. The module is single-threaded and, when in an error state, only returns an error value.

4. Roles, Services, and Authentication

There are two roles in the module (as required by FIPS 140-2) that operators may assume: a Crypto-Officer role and a User role. The Crypto-Officer and User roles are implicitly assumed by the entity accessing the services implemented by the module. No further authentication is required for a Level 1 validation. The module does not allow concurrent operators.

The following section describes the services available to each role, and each service's corresponding interface, which is depicted in Figure 1.

The module supports the following roles:

- User role – Performs cryptographic services (both in FIPS mode and non-FIPS mode), key zeroization, get status, and on-demand self-test
- Crypto-Officer role – Performs module installation and initialization

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

The User and Crypto-Officer roles are implicitly assumed by the entity accessing the module services. The module does not support a maintenance role.

4.1. Identification and Authentication

The module does not support authentication mechanisms.

4.2. Roles and Services

The module implements the required User and Crypto-Officer roles.

The following tables shows the services available in FIPS mode. The parameters and access rights are listed for each service and the associated cryptographic algorithms and roles to perform the service, cryptographic keys, or Critical Security Parameters (CSP). If the services involve the use of the cryptographic algorithms, the corresponding Cryptographic Algorithm Validation System (CAVS) certificate numbers of the cryptographic algorithms can be found in Table 7 of this security policy.

Table 4 – Cryptographic Library Services

Service	Algorithms	Role	Access	Keys/CSP
Symmetric encryption and decryption	AES, Triple-DES	User, Crypto-Officer	Read	AES, TDES key
RSA key generation	RSA, DRBG	User, Crypto-Officer	Create	RSA public-private key
RSA digital signature generation and verification	RSA	User, Crypto-Officer	Read	RSA public-private key
DSA key generation	DSA, DRBG	User, Crypto-Officer	Create	DSA public-private key
DSA domain parameter generation	DSA	User, Crypto-Officer	N/A	N/A
DSA digital signature generation and verification	DSA	User, Crypto-Officer	Read	DSA public-private key
ECDSA key generation	ECDSA, DRBG	User, Crypto-Officer	Create	ECDSA public-private key
ECDSA public key validation	ECDSA	User, Crypto-Officer	Read	ECDSA public key
ECDSA signature generation and verification	ECDSA	User, Crypto-Officer	Read	ECDSA public-private key
Random number generation	DRBG	User, Crypto-Officer	Read, Update	Entropy input string, internal state

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

Message digest	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	User, Crypto-Officer	N/A	N/A
Message authentication code (MAC)	HMAC	User, Crypto-Officer	Read	HMAC key
	CMAC with AES	User, Crypto-Officer	Read	AES key
Key encapsulation	RSA	User, Crypto-Officer	Read	RSA public-private key
Diffie-Hellman key agreement	KAS FCC (Prime curves only)	User, Crypto-Officer	Create, Read	N/A
EC Diffie-Hellman key agreement	KAS ECC (Prime curves only)	User, Crypto-Officer	Create, Read	EC Diffie-Hellman public-private keys
Keyed Hash	HMAC	User, Crypto-Officer	Create, Read	HMAC Key
Symmetric digest	AES CMAC, Triple-DES, CMAC	User, Crypto-Officer	Create, Read	N/A

Table 5 – Other FIPS-Related Services

Service	Algorithms	Role	Access	Keys/CSP
Show status	N/A	User	N/A	None
Zeroization	N/A	User	Zeroize	All CSPs
Initialize	N/A	User	N/A	None
Self-test	N/A	User, Crypto-Officer	N/A	None

5. Physical Security

This is a software module and provides no physical security.

6. Operational Environment

This module will operate in the Ubuntu 18.04 LTS Server Edition operating system, a modifiable operational environment per the FIPS 140-2 definition.

The operating system shall be restricted to a single operator mode of operation (that is, concurrent operators are explicitly excluded).

The external application that makes calls to the cryptographic module is the single user of the cryptographic module, even when the application is serving multiple clients.

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

7. Cryptographic Key Management

7.1. Cryptographic Keys and Critical Security Parameters

The module supports the Critical Security Parameters (CSP) listed in Table 6.

Table 6 – Cryptographic Module Keys and Critical Security Parameters

Name	Generation	Entry and Output	Zeroization
AES keys	Keying material can be generated with the SP 800-90A DRBG and subsequently entered into the module via API.	The key is passed into the module through API input parameters in plaintext.	EVP_CIPHER_CTX_cleanup()
HMAC key			HMAC_CTX_cleanup()
Triple-DES Keys			EVP_CIPHER_CTX_cleanup()
RSA public-private keys	The public-private keys are generated using the FIPS 186-4 Key Generation method, and the random value used in the key generation is generated using SP800-90A DRBG.	The key is passed into the module through API input parameters in plaintext. The key is passed out of the module through API output parameters in plaintext.	FIPS_rsa_free()
DSA public-private keys			FIPS_dsa_free()
ECDSA public-private keys			EC_KEY_free()
EC Diffie-Hellman public-private keys	The components used to generate the public-private keys used in EC Diffie-Hellman are generated using SP800-90A DRBG (prime curves only).	The key is passed into the module through API input parameters in plaintext. The key is passed out of the module through API output parameters in plaintext.	FIPS_dh_free()
			EC_KEY_free()
Shared secret	Generated during the Diffie-Hellman or EC Diffie-Hellman key agreement.	None	EC_KEY_free()
DRBG Seed	Derived internally as per SP 800-90A	None	FIPS_drbg_free()
Entropy input string	Obtained from NDRNG.	None	FIPS_drbg_free()
DRBG internal state (V, C, Key)	During DRBG initialization.	None	FIPS_drbg_free()

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

7.2. Random Number Generation

The module provides a Deterministic Random Bit Generator (DRBG) based on [SP800-90A] for the creation of HMAC keys, key components of asymmetric keys, symmetric keys, and internal CSPs. In addition, the module provides a Random Number Generation service to calling applications.

The DRBG supports the Hash_DRBG, HMAC_DRBG and CTR_DRBG mechanisms. The DRBG is initialized during module initialization; the module loads by default the DRBG using the CTR_DRBG mechanism with AES-256 and derivation function without prediction resistance. A different DRBG mechanism can be chosen through an API function call.

As per NIST SP 800-133, the module uses its FIPS-Approved DRBGs to generate cryptographic keys. The resulting symmetric key or generated seed is an unmodified output from the DRBG.

The module receives entropy passively from an NDRNG outside the module's logical boundary, but within the operational environment. The module receives at least 128 bits of entropy to the DRBG during initialization (seed) and reseeding (reseed). The amount of entropy depends on the security strength of the DRBG.

The module performs conditional self-tests on the output of the DRBG and NDRNG to ensure that consecutive random numbers do not repeat, and performs DRBG health tests as defined in Section 11.3 of [SP800-90A].

The module uses IG 7.14 scenario 2(b). The number of bits of entropy requested are sufficient to provide at minimum 112 bits of strength.

The AES-GCM IV generation method complies with technique #2 of IG A.5.

Note: *No assurance of the minimum strength of generated keys.*

7.3. Key Destruction/Zeroization

The memory occupied by keys is allocated by regular memory allocation operating system calls. The application is responsible for calling the appropriate zeroization function provided in the module's API. See Table 7 for more details. In addition, since all keys are stored in RAM, they can be zeroized by unloading the module from memory or rebooting the host machine.

7.4. Key Entry and Output

The module does not support manual key entry or intermediate key generation key output. The keys are provided to the module via API input parameters in plaintext form and output via API output parameters in plaintext form. This is allowed by [FIPS140-2_IG] IG 7.7, according to the "CM Software to/from App Software via GPC INT Path" entry on the Key Establishment Table.

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

7.5. Approved and Allowed Security Functions

The following table shows the CAVS certificates and their associated information of the cryptographic implementation in FIPS mode. There are algorithms, modes, and keys that have been CAVS tested but not used by the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in this table are used by the module.

Table 7 – FIPS Approved or Allowed Security Function Algorithms

Algorithm	Modes	Certificate Number

Symmetric Encryption/Decryption	AES (all variations using 128 bits, 192 bits, and 256 bits key sizes)
	• ECB – Encrypt and decrypt
	• CBC – Encrypt and decrypt
	• OFB – Encrypt and decrypt
	• CFB – Encrypt and decrypt with data segment length of 1 bit, 8 bits, and 128 bits
	• CTR – Encrypt only with an external counter source
	• CCM – Nonce length of 7-13 bytes and tag length of 4-16 bytes
	• GCM – Internal IV generation of lengths 96-1024 bits (in multiples of 8) and tag lengths of 128, 120, 112, 104, 96, 64, and 32 bits
	• Triple-DES – All variations support encrypt and decrypt using a keying option of K1, K2, K3 independent
	  ○ ECB
	  ○ CBC
	  ○ CFB – 1 bit, 8 bits, and 64 bits
	  ○ OFB
		C1011

Secure Hash Standard (SHS)	SHA-1	C1011
	SHA-2 (224, 256, 384, 512)	C1011
Data Authen-tication Code	HMAC - using SHA-1 and SHA-2 algorithms	C1011
	CMAC • Gen and Ver CMAC w/ AES 128, 192, and 256 (supports 0 length messages) • Gen and Ver CMAC w/ 3-Key TDES (supports 0 length messages)	C1011

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

Asymmetric Signature	DSA
	• PQG Gen and Ver – Using SHA-2 with probable primes p and q and unverifiable and canonical generation of g
	  ○ Gen supports mod sizes of 2048 and 3072
	  ○ Ver supports mod sizes of 1024, 2048, and 3072 with SHA-1 for mod size 1024
	• Key Pair Gen with mod sizes of 2048 and 3072
	• Sig Gen – Using SHA-2 algorithms and supports mod sizes of 2048 and 3072
	• Sig Ver – Using SHA-1 and SHA-2 algorithms and supports mode sizes of 1024, 2048, and 3072
		C1011

	ECDSA • Key Pair – Using prime curves 224, 256, 384, and 521 • PKV – Using prime curves 192, 224, 256, 384, and 521 • Sig Gen – Using prime curves 224, 256, 384, and 521 with SHA-2 algorithms • Sig Ver – Using prime curves 192, 224, 256, 384, and 521 with SHA-1 and SHA-2 algorithms	C1011
	RSA • Gen Key 9.31 supporting a random public key component e (primes p and q are random probable supporting mod sizes 2048, 3072 and 4096) • Sig Gen 9.31 – Mod sizes 2048, 3072 and 4096 with SHA-256, SHA-384, and SHA-512 • Sig Gen PKCS1.5 – Mod sizes 2048, 3072 and 4096 with SHA-224, SHA-256, SHA-384, and SHA-512 • Sig Gen PSS – Mod sizes 2048, 3072 and 4096 with SHA-224, SHA-256, SHA-384, and SHA-512 (all SHA algorithms have a SALT length of 0) • Sig Ver 9.31 – Mod sizes 1024, 2048, 3072 and 4096 with SHA-1, SHA-256, SHA-384, and SHA-512 • Sig Ver PKCS1.5 – Mod sizes 1024, 2048, 3072 and 4096 with SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512 • Sig Ver PSS – Mod sizes 1024, 2048, 3072 and 4096 with SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512	C1011
Asymmetric Key Wrapping/Encapsulation	RSA • PKCS1.5 – Mod sizes 2048 up to 16384	Allowed

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

Key Agreement (SP 800-56A rev3)	ECC - Prime curves 256, 384, and 521 FCC - Using public key size 2048 and private key size 224 or 256	KAS-SSC (vendor Affirmed)

Random Number Generation	DRBG
	• Hash – Using SHA-1 and SHA-2 with and without prediction resistance enabled
	  ○ SHA-1 has entropy input of 128 bits and a nonce of 64 bits
	  ○ SHA-224 has entropy input of 192 bits and a nonce of 96 bits
	  ○ SHA-256, SHA-384, and SHA-512 have entropy input of 256 and a nonce of 128 bits
	• HMAC – Using SHA-1 and SHA-2 with and without prediction resistance enabled
	  ○ SHA-1 has entropy input of 128 bits and a nonce of 64 bits
	  ○ SHA-224 has entropy input of 192 bits and a nonce of 96 bits
	  ○ SHA-256, SHA-384, and SHA-512 have entropy input of 256 and a nonce of 128 bits
	• CTR – Using AES 128, 192, and 256 with and without prediction resistance enabled
	  ○ AES-128 with no derivation function has entropy input of 256 bits
	  ○ AES-128 with derivation function has entropy input of 128 bits and nonce of 64 bits
	  ○ AES-192 with no derivation function has entropy input of 320 bits
	  ○ AES-192 with derivation function has entropy input of 192 bits and nonce of 128 bits
	  ○ AES-256 with no derivation function has entropy input of 384 bits
	  ○ AES-256 with derivation function has entropy input of 256 bits and nonce of 128 bits
		C1011

Cryptographic Key Generation (SP 800-133)	The resulting symmetric key or generated seed is an unmodified output from the DRBG.	CKG (Vendor Affirmed)

7.6 Non-Approved Security Functions

The following table shows the non-approved services and their associated information.

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

Table 8 – Non-Approved Security Function Algorithms

Algorithm	Modes	Certificate Number
Random Number Generation	X9.31 RNG using AES 128, 192, and 256	None
	DRBG using Dual EC	None
Symmetric Encryption/Decryption	XTS – Using AES-128 and AES-256 supporting encrypt and decrypt	None
Asymmetric Signature	RSA • GenKey9.31 – 1024 mod size with SHA-1 and SHA-2; mod sizes 2048, 3072, 4096 with SHA-1 • SigGen9.31 – 1024 mod size with SHA-1 and SHA-2; mod sizes 2048, 3072, 4096 with SHA-1 • SigGenPKCS1.5 – 1024 mod size with SHA-1 and SHA-2; mod sizes 2048, 3072, 4096 with SHA-1 • SigGenPSS – 1024 mod size with SHA-1 and SHA-2; mod sizes 2048, 3072, 4096 with SHA-1	None
	DSA • PQG Gen – 1024 mode size with SHA-1 and SHA-2; 2048 and 3072 mod sizes with SHA-1 • Key Pair Gen - 1024 mode size with SHA-1 and SHA-2; 2048 and 3072 mod sizes with SHA-1 • Sig Gen - 1024 mode size with SHA-1 and SHA-2; 2048 and 3072 mod sizes with SHA-1	None
	ECDSA • Key Pair – using prime curve 192 • Sig Gen – using prime curve 192 and prime curves 224, 256, 384, 521 using SHA-1	None

8. Self-tests

The module performs power-up self-tests and conditional self-tests.

8.1. Power-up Self-tests

The module performs power-up tests when loaded into memory, without operator intervention. Power-up tests ensure that the module is not corrupted and that the cryptographic algorithms work as expected.

While the module is executing the power-up tests, the services are not available, and input and output are inhibited. The module is not available for use by the calling application until the power-up tests are completed successfully.

If any power-up test fails, the module returns an error status and then enters an error state. The subsequent calls to the module will also fail and no further cryptographic operations are possible. If the power-up tests complete successfully, the module will return a successful status in the return code and will accept cryptographic operation service requests.

8.1.1. Cryptographic Algorithm KATs

The module performs self-tests on all FIPS-approved cryptographic algorithms supported in the Approved mode of operation, using the Known Answer Tests (KAT) and Pair-wise Consistency Tests (PCT) shown in the following table:

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

Table 9 – FIPS Algorithm Self-Tests

Algorithm	Power-Up Tests
Software Integrity	KAT HMAC SHA-256
AES	• KAT AES ECB mode with 128-bit key, encryption and decryption • KAT AES CCM mode with 192-bit key, encryption and decryption • KAT AES GCM mode with 256-bit key, encryption and decryption • KAT AES CMAC mode with 128-bit, 192-bit, and 256-bit keys, sign and verify
Triple DES	• KAT 3-key Triple-DES ECB mode, encryption and decryption • KAT 3-key Triple-DES CMAC mode, generate and verify
SHS	• KAT SHA-1 and SHA-512 • KAT SHA-224 and SHA-384 are not required for IG 9.4 • KAT SHA-256 is covered in the Integrity Test, which is allowed for IG 9.3
HMAC	• One KAT per SHA1, SHA224, SHA256, SHA384, and SHA512 • HMAC covers SHA POST requirements, which is allowed for IG 9.3
DSA	PCT DSA with L=2048, N=256, and SHA-384
ECDSA	PCT ECDSA with P-224 and SHA-512, prime curves only, keygen, sign, verify
RSA	• KAT RSA with 2048-bit key, PKCS#1 v1.5 scheme and SHA-256, signature generation • KAT RSA with 2048-bit key, PKCS#1 v1.5 scheme and SHA-256, signature verification
DRBG	• KAT Hash_DRBG without PR, with 256-bit key • KAT HMAC_DRBG without PR, with 256-bit key • KAT CTR_DRBG without PR, with DF, with 256-bit key • KAT CTR_DRBG without PR, without DF, with 256-bit key
Diffie-Hellman	Primitive “Z” Computation KAT with 2048-bit key
EC Diffie-Hellman	KAT shared secret calculation

8.1.2. Software and Firmware Integrity Tests

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

8.2. Conditional Self-tests

The module performs conditional tests on the cryptographic algorithms, using Pair-wise Consistency Tests

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.

(PCT) and Continuous Random Number Generator Tests (CRNGT), as shown in Table 10.

Table 10 – Conditional Tests

Algorithm	Conditional Tests
DSA key generation	PCT, signature generation and verification
ECDSA key generation	PCT, signature generation and verification
RSA key generation	• PCT, signature generation and verification • PCT for encryption and decryption
DRBG	• Tested as required by SP800-90, Section 11 • FIPS 140-2 continuous test for stuck fault
NDRNG	• FIPS 140-2 continuous test for stuck fault

9. Crypto-Officer and User Guidance

9.1. Secure Setup and Initialization

The module is installed as part of the Stealth Linux endpoint package for Ubuntu 18.04 LTS. The endpoint package can be installed on an Ubuntu system or on a Unisys network appliance. All versions of the Stealth Linux endpoint package for Ubuntu 18.04 LTS use the version of the module covered by this Security Policy.

9.2. Module Security Policy Rules

When operating in a FIPS Approved Mode, the user or administrator shall not use any debugging or tracing software to inspect the module.

As per FIPS IG A.13, if Triple-DES is employed, the user or administrator is responsible for ensuring that the module limits the use of any single Triple-DES key to less than 2^16 encryptions before the key is changed.

10. Mitigation of Other Attacks

The module does not mitigate against any specific attacks.

© Copyright 2021 Unisys Corporation
This document may be freely reproduced and distributed whole and intact including this Copyright Notice.