Senetas Corporation Ltd., distributed by Thales SA

CE Crypto Module

Module Version: 5.5.0

FIPS 140-3 Non-Proprietary Security Policy

Level 1 Validation

September 2024

www.senetas.com

Document History

Authors	Date	Version	Comment
Senetas Corp. Ltd.	22-Dec-2023	1.00	CMVP Release for firmware version 5.5.0
Senetas Corp. Ltd.	04-Sep-2024	1.01	Interim validation update

Table of Contents

1. General

This is a non-proprietary FIPS 140-3 Security Policy for the Senetas Corporation Ltd. CE Crypto Module v.5.5.0. This Security Policy specifies the security rules under which the module operates to meet the FIPS 140-3 Level 1 requirements.

The CE Crypto Module is used in a range of Senetas encryption appliances. The vendor distributes under their own Senetas brand, and jointly with their master worldwide distributor, Thales SA.

FIPS 140-3 (Federal Information Processing Standards Publication 140-3), *Security Requirements for Cryptographic Modules*, specifies the security requirements for a cryptographic module utilized within a security system protecting sensitive but unclassified information. Based on four security levels for cryptographic modules, this standard identifies requirements in twelve sections. For more information about the NIST/CCCS Cryptographic Module Validation Program (CMVP) and the FIPS 140-3 standard, visit www.nist.gov/cmvp .

This Security Policy, using the terminology contained in the FIPS 140-3 specification, describes how the CE Crypto Module complies with the twelve sections of the standard. In this document, the CE Crypto Module is more generally referred to as “the module”.

This Security Policy contains only non-proprietary information. Any other documentation associated with FIPS 140-3 conformance testing and validation is proprietary and confidential to Senetas Corporation Ltd. and is releasable only under appropriate non-disclosure agreements. For more information describing the module and associated platforms, visit http://www.senetas.com.

1.1 References

For more information on the FIPS 140-3 standard and validation program please refer to the National Institute of Standards and Technology website at www.nist.gov/cmvp.

The following standards from NIST are all available via the URL: www.nist.gov/cmvp .

[1] *FIPS PUB 140-3: Security Requirements for Cryptographic Modules.*

[2] *NIST Special Publication (SP) 800-140 FIPS 140-3 Derived Test Requirements (DTR).*

[3] *NIST Special Publication (SP) 800-140A CMVP Documentation Requirements.*

[4] *NIST Special Publication (SP) 800-140B CMVP Security Policy Requirements.*

[5] *NIST Special Publication (SP) 800-140Crev2 CMVP Approved Security Functions.*

[6] *NIST Special Publication (SP) 800-140Drev2 CMVP Approved Sensitive Security Parameter Generation and Establishment Methods.*

[7] *NIST Special Publication (SP) 800-140E CMVP Approved Authentication Mechanisms.*

[8] *NIST Special Publication (SP) 800-140F CMVP Approved Non-Invasive Attack Mitigation Test Metrics.*

[9] *ISO/IEC 19790:2012(E), Information technology — Security techniques — Security requirements for cryptographic modules.*

[10] *ISO/IEC 24759:2017(E), Information technology — Security techniques — Test requirements for cryptographic modules.*

[11] *NIST Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program.*

[12] *Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197.*

[13] *Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4.*

[14] *Secure Hash Standard (SHS), Federal Information Processing Standards Publication 180-4.*

[15] *NIST Special Publication (SP) 800-131Arev2, Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths.*

[16] *NIST Special Publication (SP) 800-90Arev1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators.*

[17] *NIST Special Publication (SP) 800-56Arev3 Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography.*

[18] *Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4.*

[19] *NIST Special Publication (SP) 800-56Brev2, Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography.*

[20] *NIST Special Publication (SP) 800-108rev1 Recommendation for Key Derivation Using Pseudorandom Functions.*

[21] *NIST Special Publication (SP) 800-56Crev2 Recommendation for Key-Derivation Methods in Key Establishment Schemes.*

[22] *NIST Special Publication (SP) 800-90B, Recommendation for the Entropy Sources Used for Random Bit Generation.*

[23] *NIST Special Publication (SP) 800-133rev2, Recommendation for Cryptographic Key Generation.*

[24] *NIST Special Publication (SP) 800-67rev2, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher.*

[25] *NIST Special Publication (SP) 800-135rev1, Recommendation for Existing Application-Specific Key Derivation Functions*

1.2 Acronyms and Abbreviations

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
API	Application Programming Interface
CBC	Cipher Block Chaining
CCCS	Canadian Centre for Cyber Security
CFB	Cipher Feedback
CAVP	Cryptographic Algorithm Validation Program
CMVP	Cryptographic Module Validation Program
CNF	Cloud Network Function
CSP	Critical Security Parameter
CTR	Counter Mode
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ESV(NP)	Non-Physical Entropy Source
ESV	Entropy Source Validation
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode
HMAC	Keyed-Hash Message Authentication Code
IV	Initialisation Vector
KAS-ECC	Elliptic Curve Key Agreement Scheme (ECDH)
KAS-FCC	Finite Field Key Agreement Scheme (DH)
KAT	Known Answer Test
KDF	Key Derivation Function
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OAEP	Optimal Asymmetric Encryption Padding
PAA	Processor Algorithm Accelerator
PKCS	Public Key Cryptography Standards
PSP	Public Security Parameter
PUB	Publication
RAM	Random Access Memory
RNG	Random Number Generator
RSA	Rivest Shamir and Adleman Public Key Algorithm
SP	Special Publication
SHA	Secure Hash Algorithm
SSP	Sensitive Security Parameter
TOEPP	Tested Operational Environment Physical Perimeter
VNF	Virtual Network Function

1.3 Security Levels

The module meets the overall Security Level 1 requirements for FIPS 140-3. See Table 1 below, which indicates the security level of each of the twelve sections of the FIPS 140-3 standard.

Table 1 Security Levels

ISO/IEC 24759 Section 6 [Number Below]	FIPS 140-3 Section Title	Security Level
1	General	1
2	Cryptographic Module Specification	1
3	Cryptographic Module Interfaces	1
4	Roles, Services and Authentication	1
5	Software/Firmware Security	1
6	Operational Environment	1
7	Physical Security	1
8	Non-invasive Security	N/A
9	Sensitive Security Parameter Management	1
10	Self-tests	1
11	Life Cycle Assurance	1
12	Mitigation of Other Attacks	N/A

2. Cryptographic Module Specification

The CE Crypto Module version 5.5.0 is a firmware cryptographic module running on a multi-chip standalone general-purpose compute platform. The module provides low-level cryptographic primitives to the overall platform and its functions.

The Module exists as a number of shared libraries and is linked against various encryption applications to supply all cryptographic operations as required by those applications.

2.1 Operational Environment

The module has been tested by the certification lab, Lightship Security, Inc. on the following platform with and without PAA:

Table 2 Tested Operational Environments

#	Operating System	Hardware Platform	Processor	PAA/Acceleration
1	Debian Linux v11	Dell VEP4600	Intel Xeon D-2145NT (Skylake)	AES-NI

In addition to the platforms listed in Table 2 above, Senetas Corporation has also tested the module on the following platforms and claims vendor affirmation on them:

Table 3 Vendor Affirmed Operational Environments

#	Operating System	Hardware Platform
1	Debian Linux v11	Dell VEL1485 with Intel Atom C3000 (Goldmont) CPU
2	Debian Linux v11	CONTEC CPS-BXC-200 with Intel Atom x7-E3950 CPU
3	Debian Linux v11 (VNF/CNF)	Dell VEP4600 with Intel Xeon D-2145NT (Skylake) CPU

Note: The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when ported onto an Operating Environment that is not listed on the validation certificate.

2.2 Modes of Operation

The module only supports an approved mode of operation.

2.3 Cryptographic Algorithms

2.3.1 Approved Algorithms

Table 4 lists the approved security functions of the module in the approved mode of operation. There are algorithms, modes, and key/moduli sizes that have been CAVP-tested but are not used by any approved service of the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in this table are used by an approved service of the module.

Table 4 Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode/Method	Description/ Key Size(s)/ Key Strength(s)	Use/ Function
A4648	AES FIPS PUB 197, SP 800-38A SP 800-38D	CFB128 (e/d; 128,256) CTR (e; 128, 256) ECB[1] (e/d; 128, 256) CBC (e/d; 128,256) GCM (e/d; 128,256 Internal IV, AAD=0 to 256)	128-bit 256-bit	Symmetric Encryption and Decryption
A4648	RSA FIPS186-4	KeyGen; MOD: 2048 ALG[RSASSA-PKCS1_V1_5]; SigGen; MOD: 2048 SHS: SHA-256 SigVer; MOD: 2048 SHS: SHA-256, SHA-384 and SHA-512 SigVer; MOD: 4096 SHS: SHA-256, SHA-384 and SHA-512	2048-bit 4096-bit	Asymmetric Key Generation, Digital Signature Generation and Verification
A4648	ECDSA FIPS186-4	KeyGen KeyVer SigGen SigVer	P-256 P-384 P-521	Asymmetric Key Generation, Digital Signature Generation and Verification
A4648	KAS-ECC SP 800-56Arev3	(Cofactor) Ephemeral Unified Model key agreement	NIST P-256, P-384 and P-521 curves are supported and SHA-256, SHA-384 and SHA-512 (respectively) are used for key derivation Key establishment methodology provides between 128 and 256 bits of encryption strength	Key Agreement
A4648	KAS-FFC SP 800-56Arev3	dhEphem key agreement	MODP-2048 bit Oakley Group 14 using SHA-256 for key derivation Key establishment methodology provides 112 bits of encryption strength	Key Agreement
A4648	SHA FIPS 180-4	SHA-1 (BYTE only) SHA-256 (BYTE only) SHA-384 (BYTE only) SHA-512 (BYTE only)		Message Digest
A4648	HMAC	HMAC-SHA-1	Key Sizes Ranges Tested: KS<BS	Message Authentication

	FIPS 198-1	HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512		
A4648	DRBG SP 800-90Arev1	Hash_Based DRBG: [Prediction Resistance Tested: Not Enabled (SHA-256)]		Random Number Generation
A4648	KBKDF SP 800-108rev1	Counter based KDF using HMAC-SHA-256		Key Derivation
A4648	KTS-IFC SP 800-56Brev2 FIPS 140-3 IG D.G	RSA-OAEP-256 Key Transport rsakpg1-basic	2048-bit Key establishment methodology provides 112 bits of encryption strength	Key Encapsulation/ Un-encapsulation
A4648	KTS FIPS 140-3 IG D.G	AES-256 CFB key wrapping authenticated with HMAC-SHA-256	256-bit Key establishment methodology provides 256 bits of encryption strength	Key Wrapping and Unwrapping
A3449	SHA3 FIPS 202	SHA3-256 (BYTE only)	256-bit	ESV Conditioning
E49	ESV (NP) SP 800-90B		256-bit	Entropy Source for DRBG
Vendor Affirmed	CKG[2] SP 800-133rev2	Sections 5.1 & 5.2 - Asymmetric key generation using unmodified DRBG output		Key Generation

Note 1: AES-ECB Is only validated as part of the AES-CTR validation. The mode is not actively used by the module.
Note 2: The seeds used in the asymmetric key generation are from the unmodified output from the Approved NIST SP 800-90A DRBG.

The module does not implement:

- Non-Approved Algorithms Allowed in the Approved Mode of Operation.
- Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed.
- Non-Approved Algorithms Not Allowed in the Approved Mode of Operation.

2.3.1.1 TLS AES-GCM Key and IV generation (refer to Table 4 above)

Whilst the module does not provide the TLS protocol (this protocol has not been reviewed or tested by the CAVP and CMVP) itself, it does supply the underlying cryptographic functionality required by TLS including AES-GCM. IG C.H Scenario 1a applies:

- The module conforms to TLSv1.2 GCM cipher suites as specified in SP 800-52rev2, Section 3.3.1.
- When the nonce_explicit part of the IV exhausts the maximum number of possible values for a given session key, the module will trigger a handshake to establish a new encryption key according to RFC 5246.
- In case the module's power is lost and then restored, a new key for use with the AES-GCM encryption/decryption shall be established.

2.4 Cryptographic Boundary

The CE Crypto Module is a firmware library, providing user space based cryptographic primitives for use by the wider system. The cryptographic boundary is depicted in the diagram below.

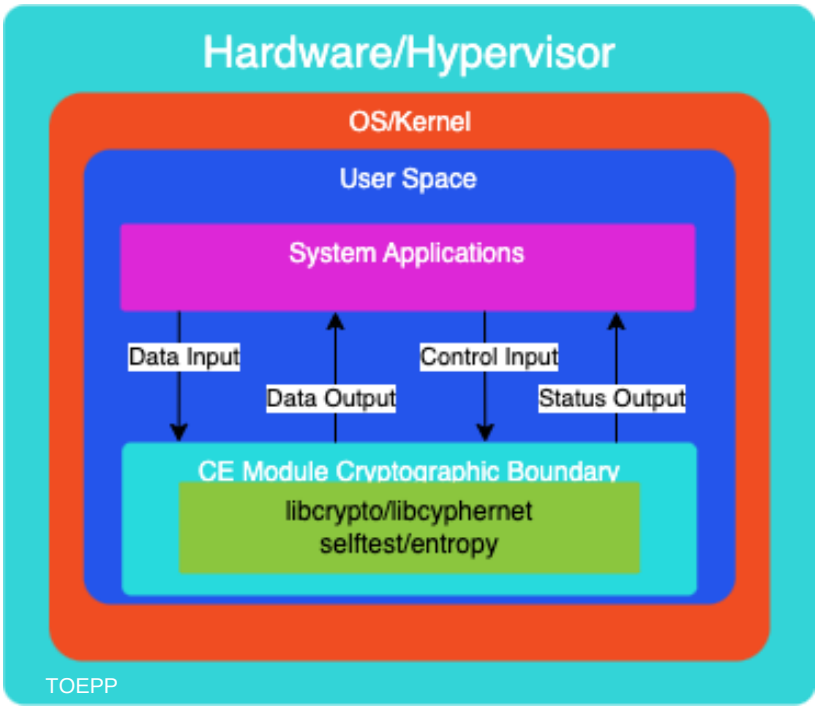

Figure 1 Cryptographic Boundary Block Diagram

3. Cryptographic Module Interfaces

As a firmware only module, the module does not have any physical ports. Any reference to physical ports refers to that hardware on which the module is operating and outside of the cryptographic boundary.

With regard to logical interfaces, the cryptographic API (C programming language) delineates the module interfaces.

Table 5 Ports and Interfaces

Logical Interface	Data that passes over port/interface
Data Input	Data read from variables passed in the API
Data Output	Data written to user supplied variables or pointers in the API
Control Input	The API function called and the parameters by which it is invoked.
Status Output	The return value of the invoked API call.
Power Input	

Note: The Control Output interface is not applicable.

4. Roles, Services and Authentication

The cryptographic module supports a single role of Crypto Officer. No authentication mechanism is provided, this aligns with the requirements for a FIPS 140-3 Level 1 module. The Crypto Officer has access to all approved services.

The “Roles and Authentication” table listed in SP 800-140B is not applicable.

4.1 Supported Roles

The supported role and services are summarized in Table 6

Table 6 Roles, Service Commands, Input and Output

Role	Service	Input	Output
Crypto Officer	AES Encryption/ Decryption	API call parameters, AES Keys and cipher/plain text	Status, Cipher/Plain text
	RSA Key Generation	API call parameters	Status, RSA Private and RSA Public Keys
	RSA Signature Generation and Verification	API call parameters, RSA Private and RSA Public Keys	Status, Signature
	ECDSA Key Generation	API call parameters	Status, ECDSA Private and ECDSA Public Keys
	ECDSA Signature Generation and Verification	API call parameters, ECDSA Private and ECDSA Public Keys	Status, Signature
	ECDH Key Agreement	API call parameters	Status, Agreed Key
	DH Key Agreement	API call parameters	Status, Agreed Key
	Secure Hash Generation	API call parameters, message	Status, Hash
	HMAC Generation and Verification	API call parameters, HMAC Key and message	Status, Hash
	Random Number Generation	API call parameters	Random numbers
	Key Based Key Derivation Function	API call parameters, KBKDF Key Derivation Key	Status, Derived Key
	RSA Key Encapsulation/ Un-encapsulation	API call parameters, RSA Public and RSA Private Keys, AES key to be encapsulated	Status, Symmetric Key
	AES Key Wrapping	API call parameters, AES Key-wrapping Key and AES Key to be wrapped	Status, Symmetric Key
	Self-test	API call parameters	Test results
	Show Status	API call parameters	Status
	Show Module Info	API call (inventory)	Module version number and description
	Zeroisation	Reboot command	N/A

4.2 Roles and Services

The CE Crypto Module supports the Crypto Officer services listed in the following table.

Legend for access rights column in Table 7:

G = Generate: The module generates or derives the SSP.

R = Read: The SSP is read from the module (e.g. the SSP is output).

W = Write: The SSP is updated, imported, or written to the module.

E = Execute: The module uses the SSP in performing a cryptographic operation.

Z = Zeroise: The module zeroises the SSP.

N/A - Not Applicable.

4.2.1 Approved Services

The module supports the approved services listed in Table 7.

Table 7 Approved Services

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys or SSPs	Indicator
AES Encryption/ Decryption		CFB128 (e/d; 128,256) CTR (e; 128, 256) ECB (e/d; 128, 256) CBC (e/d; 128,256) GCM (e/d; 128,256 Internal IV, AAD=0 to 256)	AES Keys	CO	W, E	Status
RSA Key Generation		KeyGen; MOD: 2048 CKG	RSA Private Keys RSA Public Keys	CO	G, R, E	Status
RSA Signature Generation and Verification		ALG[RSASSA-PKCS1_V1_5]; SigGen; MOD: 2048 SHS: SHA-256 SigVer; MOD: 2048 SHS: SHA-256, SHA-384 and SHA-512 SigVer; MOD: 4096 SHS: SHA-256, SHA-384 and SHA-512	RSA Private Keys RSA Public Keys	CO	W, E	Status
ECDSA Key Generation		KeyGen KeyVer CKG	ECDSA Private Keys ECDSA Public Keys	CO	G, R, E	Status
ECDSA Signature Generation and Verification		SigGen SigVer	ECDSA Private Keys ECDSA Public Keys	CO	W, E	Status
ECDH Key Agreement		CKG KAS-ECC: (Cofactor) Ephemeral Unified Model key agreement	ECDHE Private Keys ECDHE Public Keys ECDHE Shared Secret	CO	G, R, W, E	Status
DH Key Agreement		CKG KAS-FFC: dhEphem key agreement	Diffie Hellman Private Keys Diffie Hellman Public Keys Diffie Hellman Shared Secret	CO	G, R, W, E	Status
Secure Hash Generation		SHA-1 (BYTE only) SHA-256 (BYTE only) SHA-384 (BYTE only) SHA-512 (BYTE only)	None	CO	N/A	Status
HMAC Generation		HMAC-SHA-1	HMAC Key	CO	W, E	Status

and Verification		HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512				
Random Number Generation		Hash_Based DRBG: [Prediction Resistance Tested: Not Enabled (SHA-256)]	DRBG Entropy Input and Nonce DRBG Seed DRBG V and C internal state parameters	CO	G, R, E	Status
Key Based Key Derivation Function		Counter based KDF using HMAC-SHA-256	KBKDF Key Derivation Key Derived AES Key-wrapping Key HMAC Key	CO	G, R, W, E	Status
RSA Key Encapsulation/ Un-encapsulation		RSA-OAEP-256 Key Transport	RSA Private Keys RSA Public Keys	CO	W, E	Status
AES Key Wrapping/ Unwrapping		AES-256 CFB key wrapping authenticated with HMAC-SHA-256	AES Key-wrapping Key	CO	W, E	Status
Self-test	Run self-tests	N/A	None	CO	N/A	Status
Show Status	API call return code	NA	None	CO	N/A	None
Show Module Info	API call (inventory), module version number and description	NA	None	CO	N/A	None
Zeroisation	Reboot Module	NA	All	CO	Z	None

5. Software/Firmware Security

5.1 Software/Firmware Integrity Test

The approved SHA-256 algorithm implemented in the module is used to verify the integrity of the module. If this integrity test fails, the module is prevented from providing any cryptographic services and the module is effectively disabled.

Refer to Section 10.1 for more detail.

5.1.1 On Demand Software/Firmware Integrity Test

On demand testing can be initiated by rebooting the module's host platform.

6. Operational Environment

The module is designed to operate as a component of a larger general-purpose operating system. The operational environment is non-modifiable.

7. Physical Security

The module is a firmware module with a multi-chip standalone cryptographic embodiment. The module's host platform provides production-grade components and chassis, using standard passivation.

8. Non-Invasive Security

The requirements in this section are Not Applicable.

9. Sensitive Security Parameter Management

9.1 Cryptographic Keys and SSPs

The following table identifies the Cryptographic Keys and Sensitive Security Parameters (SSPs) employed within the module.

Table 8 SSPs

Key/SSP Name/ Type	Strength	Security Function and Cert. Number	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use & Related Keys
AES Keys (CSP)	128-bit / 256-bit	AES A4648	N/A	Imported from calling application	N/A	In volatile system memory by caller (plaintext)	• Internal buffers cleared. • Power cycle	Symmetric Encryption and Decryption
RSA Private Keys (CSP)	2048-bits	RSA A4648 KTS-IFC A4648	Internal FIPS 186-4 SP 800-133rev2 Key Generation using SP 800-90Arev1 DRBG	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	• Internal buffers cleared • Power cycle	Key Generation, Digital Signature/ Generation, Key un-encapsulation
RSA Public Keys (PSP)	2048-bits	RSA A4648 KTS-IFC A4648	Internal FIPS 186-4 SP 800-133rev2 Key Generation using SP 800-90Arev1 DRBG	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	• Internal buffers cleared • Power cycle	Key Generation, Digital Signature/ Verification, Key encapsulation
ECDSA Private Keys (CSP)	P-256 P-384 P-521	ECDSA A4648	Internal FIPS 186-4 SP 800-133rev2 Key Generation using SP 800-90Arev1 DRBG	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	• Internal buffers cleared • Power Cycle	Key Generation, Digital Signature/ Generation
ECDSA Public Keys (PSP)	P-256 P-384 P-521	ECDSA A4648	Internal FIPS 186-4 SP 800-133rev2 Key Generation using SP 800-90Arev1 DRBG	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	• Internal buffers cleared • Power Cycle	Key Generation, Digital Signature/ Verification

ECDHE Private Keys (CSP)	P-256 P-384 P-521	KAS ECC A4648	Internal SP 800-56Arev3 SP 800-133rev2 Key Generation using SP 800-90Arev1 DRBG	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	Power cycle	KAS-ECC
ECDHE Public Keys (PSP)	P-256 P-384 P-521	KAS ECC A4648	Internal SP 800-56Arev3 SP 800-133rev2 Key Generation using SP 800-90Arev1 DRBG	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	Power cycle	KAS-ECC
ECDHE Shared Secret (CSP)	P-256 P-384 P-521	KAS ECC A4648		Imported from and exported to calling function	Computed during the SP 800-56Arev3 compliant ECDH key agreement	In volatile system memory by caller (plaintext)	Power cycle	KAS-ECC
Diffie Hellman Private Keys (CSP)	2048-bits	KAS-FFC A4648	Internal SP 800-56Arev3 SP 800-133rev2 Key Generation using SP 800-90Arev1 DRBG	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	Power cycle	KAS-FCC
Diffie Hellman Public Keys (PSP)	2048-bits	KAS-FFC A4648	Internal SP 800-56Arev3 SP 800-133rev2 Key Generation using SP 800-90Arev1 DRBG	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	Power cycle	KAS-FCC
Diffie Hellman Shared Secret (CSP)	2048-bits	KAS-FFC A4648		Imported from and exported to calling function	Computed during the SP 800-56Arev3 compliant ECDH key agreement	In volatile system memory by caller (plaintext)	Power cycle	KAS-FCC

DRBG Seed (CSP)	440-bit	DRBG A4648	Internal	N/A	N/A	In volatile system memory by caller (plaintext)	Power cycle	Used for SP 800-90rev1 Hash_DRBG the 440-bit seed (initial V or state) value internally generated from nonce along with entropy input. A software based non-deterministic RNG is used for seeding the approved SP 800-90Arev1 DRBG.
DRBG Entropy Input and Nonce (CSP)		DRBG A4648	Internal from ESV (NP)	N/A	N/A	In volatile system memory by caller (plaintext)	Power cycle	Used for SP 800-90rev1 Hash_DRBG as input to the instantiate function.
DRBG V and C internal state parameters (CSP)		DRBG A4648	Internal	N/A	N/A	In volatile system memory by caller (plaintext)	Power cycle	The V and C parameters store the internal state of the SP 800-90rev1 DRBG.
KBKDF Key Derivation Key (CSP)	256-bits	KBKDF A4648	Internal SP 800-133rev2 Key Generation using SP 800-90Arev1 DRBG	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	Power Cycle	Key Derivation
HMAC Key (CSP)	256-bits	HMAC A4648	Internal Derived from KBKDF Key Derivation Key using SP 800-108 compliant KDF	Imported from and exported to calling function	N/A	In volatile system memory by caller (plaintext)	Power Cycle	Message Authentication
AES Key-wrapping Key (CSP)	256-bits	KTS A4648	Internal Derived from KBKDF Key Derivation Key using SP 800-108 compliant KDF	Imported from and exported to calling application	N/A	In volatile system memory by caller (plaintext)	Power Cycle	Key Transport

9.2 Random Number Generation/Entropy

An approved NIST [SP800-90A] deterministic random bit generator using a hash based DRBG (SHA-256) is used.

The DRBG is seeded via a Linux Inter Process Communication (IPC) pipe (/tmp/sp80090bd), which in turn is filled via a user space daemon that utilises the software-based CPU jitter library (https://www.chronox.de/jent.html).

The user space daemon ensures a watermark entropy pool is maintained for seeding the DRBG.

Based on testing and analysis, the estimated minimum amount of entropy per output bit is 1.0 bits. The overall amount of generated entropy meets the required security strength of 256 bits based on the entropy per bit and the amount of entropy requested by the module.

Table 9 Non-Deterministic Random Number Generation Specification

Entropy Sources	Minimum number of bits of entropy	Details
Senetas CPU Jitter Entropy Source **ESV E49**	256 bits	The module employs a software based random bit generator

10. Self-tests

The module performs pre-operational self-tests and conditional self-tests to assure that faults have not been introduced that would prevent the module's correct operation.

10.1 Pre-Operational Self-Tests

The module performs a Pre-Operational Software/Firmware Integrity Test.

10.2 Conditional Self-tests

The module performs a set of Conditional Cryptographic Algorithm Self-Tests. These Conditional Cryptographic Algorithm Self-Tests run in the pre-operational state.

The cryptographic algorithm used to perform the approved integrity technique for the Pre-Operational Software/Firmware Integrity Test (listed in Table 10, below), is tested using a Cryptographic Algorithm Self-Test (CAST) prior to the Pre-Operational Software/Firmware Integrity Test.

A Conditional Pair-wise Consistency Test is performed on asymmetric key pairs generated by the module (refer to Table 10 below).

10.3 On-Demand and Periodic Self-tests

The Crypto Officer can initiate the Pre-Operational Self-Test and Conditional Cryptographic Algorithm Self-Tests on-demand and for periodic testing of the module by issuing a reboot of the module's host operating system.

10.4 Error State

Failure of the Pre-Operational Self-Test or any of the Conditional Cryptographic Algorithm Self-Tests will cause the module to remain in the pre-operational state (or error state). Once all of the self-tests have passed the pre-operational flag is removed and the module will transition to the operational state.

The self-tests are detailed in Table 10.

Table 10 Self-Tests

Pre-Operational Self-Tests	Notes
Pre-Operational Software/Firmware Integrity Test	
SHA-256	
Conditional Self-Tests	Notes
Cryptographic Algorithm Self-Tests	
SHA-1, SHA-256, SHA-384, SHA-512	KATs
HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	KATs
KDF CTR HMAC-SHA-256	KAT
AES-CFB128-128 (e/d), AES-CFB128-256 (e/d)	KATs
AES-CBC-128 (e/d), AES-CBC-256 (e/d)	KATs
AES-GCM-128 (e/d), AES-GCM-256 (e/d)	KATs
RSA-2048 (priv enc, pub dec)	KATs
RSA-2048 (pub enc, priv dec)	KATs
RSA-4096 (priv enc, pub dec)	KATs
RSA-4096 (pub enc, priv dec)	KATs
RSA-2048-OAEP-SHA2 (pub enc, priv dec)	KATs
RSA-2048 sign/verify	KATs
RSA-4096 sign/verify	KATs
SP 800-90Arev1 HASH-DRBG Instantiate, reseed, generate, un-instantiate	KATs
DH dhEphem 2048 MODP group SP 800-56Arev3	KAT
ECDH (Cofactor) Ephemeral Unified Model SP 800-56Arev3	KAT

ECDH P-256, P-384, P-521 (primitive KAT)	KATs
ECDSA P-256-SHA256, P-384-SHA384, P-521-SHA512 sign/verify	KATs
Entropy Related Health Tests	The entropy source is tested using adaptive proportion and repeat count tests compliant with SP 800-90B Section 4.4 during the start-up sequence and then continuously.
Conditional Pair-wise Consistency	RSA Public and Private keys are used for the calculation and verification of digital signatures and for key transport. These keys are tested for consistency, based on their purpose, at the time they are used. RSA wrapping keys are tested by an encrypt/ decrypt pair-wise consistency test; signature keys are tested by a sign/verify pair-wise consistency test. ECDSA Public and Private keys are used for the calculation and verification of digital signatures. These keys are tested at the time they are used with a sign/verify pair-wise consistency test. ECDH Public and Private keys are used for SP 800-56Arev3 approved key agreement. These keys are tested at the time they are used with a pair-wise consistency test. DH Public and Private keys are used for SP 800-56Arev3 approved key agreement. These keys are tested at the time they are used with a pair-wise consistency test.

11. Life-cycle Assurance

The module is part of a larger Senetas encryption platform distributed on a range of comparable compute devices as a complete Linux distribution and set of services, and as such is installed as part of the encompassing Senetas encryption application.

The development and operational processes around which the module is supported are strictly controlled across the complete development life cycle and supply chain and externally audited for correctness.

12. Mitigation of Other Attacks

The requirements in this section are Not Applicable.