Senetas Corporation Ltd, distributed by Thales SA
CE Crypto Module

FIPS 140-3 Non-Proprietary Security Policy
Document Version 1.00
17 July 2026

Table of Contents

List of Tables

List of Figures

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) 800-140Brev1 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*

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 General

1.1 Overview

This is a non-proprietary FIPS 140-3 Security Policy for the Senetas Corporation Ltd. CE Crypto Module v5.5.0. 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 https://www.senetas.com.

1.2 Security Levels

The module meets the overall Security Level 1 requirements for FIPS 140-3. The table below reflects the individual security areas of FIPS 140-3, as well as the Security Levels of those individual areas.

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
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

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.

Module Type: Firmware

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary and Tested Operational Environment's Physical Perimeter (TOEPP):

The CE Crypto Module cryptographic boundary and TOEPP are depicted in the diagram below. The cryptographic boundary encompasses the libcypernet, libcrypto, entropy and selftest libraries and is defined by the innermost cyan block in the diagram. The TOEPP encompasses the host general-purpose computing platform that the module is installed on and is defined by the outermost cyan block in the diagram.

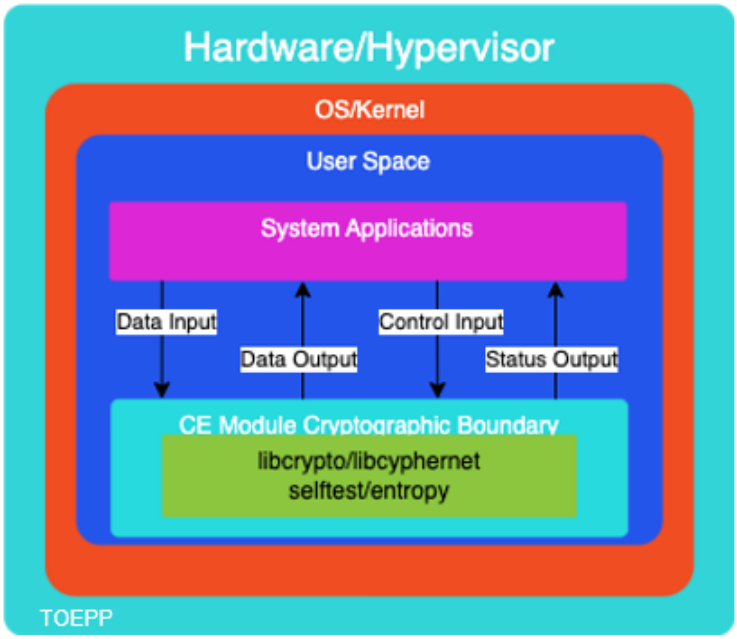

Figure 1 – Cryptographic Boundary Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
libcyphernet.so, libcrypto.so, selftest, libjitterentropy.so	5.5.0		SHA2-256

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

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Debian Linux v11	Dell VEP4600	Intel Xeon D-2145NT (Skylake)	Yes		5.5.0
Debian Linux v11	Dell VEP4600	Intel Xeon D-2145NT (Skylake)	No		5.5.0

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

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

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

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

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

2.3 Excluded Components

None

2.4 Modes of Operation

Modes List and Description:

The module only supports an approved mode of operation.

Mode Name	Description	Type	Status Indicator
FIPS mode	The module only supports an approved mode of operation.	Approved	Pre-Operational flag reset (Global)

Table 5: Modes List and Description

2.5 Algorithms

Approved Algorithms:

The module implements the approved algorithms in the table below.

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4648	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-CFB128	A4648	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-CTR	A4648	Direction - Encrypt Key Length - 128, 256 Payload Length - Payload Length: 128 Counter Tests Performed - Yes	SP 800-38A
AES-GCM	A4648	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.2 Key Length - 128, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 Payload Length - Payload Length: 120, 128, 248, 256, 0 AAD Length - AAD Length: 0, 120, 128, 248, 256	SP 800-38D
ECDSA KeyGen (FIPS186-4)	A4648	Curve - P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4648	Curve - P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4648	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4648	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
Hash DRBG	A4648	Prediction Resistance - No Supports Reseed - No Mode - SHA2-256 Entropy Input - Entropy Input: 512 Nonce - Nonce: 128 Personalization String Length - Personalization String Length: 0-1024 Increment 8 Additional Input - Additional Input: 0-1024 Increment 8 Returned Bits - 1024	SP 800-90A Rev. 1
HMAC-SHA-1	A4648	MAC - MAC: 160 Key Length - Key Length: 80-320 Increment 8	FIPS 198-1
HMAC-SHA2-256	A4648	MAC - MAC: 256 Key Length - Key Length: 80-320 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4648	MAC - MAC: 384 Key Length - Key Length: 80-320 Increment 8	FIPS 198-1
HMAC-SHA2-512	A4648	MAC - MAC: 512 Key Length - Key Length: 80-320 Increment 8	FIPS 198-1
KAS-ECC Sp800-56Ar3	A4648	Domain Parameter Generation Methods - P-256, P-384, P-521 Function - Full Validation, Key Pair Generation iutId - 123456ABCD Scheme - ephemeralUnified - KAS Role - Initiator, Responder KDF Methods - oneStepKdf - Auxiliary Function Methods - Auxiliary Function Name - SHA2-256 Fixed Info Pattern - literal[abcd1234]||uPartyInfo||vPartyInfo	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
		Fixed Info Encoding - Concatenation Key Length - 256	
KAS-FFC Sp800-56Ar3	A4648	Domain Parameter Generation Methods - MODP-2048 Function - Full Validation, Key Pair Generation iutId - 123456ABCD Scheme - dhEphem - KAS Role - Initiator, Responder KDF Methods - oneStepKdf - Auxiliary Function Methods - Auxiliary Function Name - SHA2-256 Fixed Info Pattern - literal[abcd1234]||uPartyInfo||vPartyInfo Fixed Info Encoding - Concatenation Key Length - 256	SP 800-56A Rev. 3
KDF SP800-108	A4648	KDF Mode - Counter MAC Mode - HMAC-SHA2-256 Supported Lengths - Supported Lengths: 256 Fixed Data Order - Before Fixed Data Counter Length - 32 Supports Empty IV - No Requires Empty IV - No Custom Key In Length - 0	SP 800-108 Rev. 1
KTS-IFC	A4648	Function - keyPairGen, partialVal IUT ID - CAFECAFE Modulo - 2048 Key Generation Methods - rsakpg1-basic Fixed Public Exponent - 010001 Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Hash Algorithms - SHA2-256 Supports Null Associated Data - Yes Associated Data Encoding - concatenation Key Length - 256	SP 800-56B Rev. 2
RSA KeyGen (FIPS186-4)	A4648	Key Generation Mode - B.3.6 Modulo - 2048 Primality Tests - Table C.3 Info Generated By Server - No Public Exponent Mode - Fixed Fixed Public Exponent - 010001 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A4648	Signature Type - PKCS 1.5 Modulo - 2048 Hash Pair - Hash Algorithm - SHA2-256	FIPS 186-4
RSA SigVer (FIPS186-4)	A4648	Signature Type - PKCS 1.5 Modulo - 2048, 4096 Hash Pair - Hash Algorithm - SHA2-256 Public Exponent Mode - Fixed Fixed Public Exponent - 010001	FIPS 186-4
Safe Primes Key Generation	A4648	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
Safe Primes Key Verification	A4648	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3
SHA-1	A4648	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4
SHA2-256	A4648	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4
SHA2-384	A4648	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A4648	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA3-256	A3449	Message Length - Message Length: 0-65536 Increment 8	FIPS 202

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

The module implements the vendor-affirmed algorithms in the table below.

Name	Properties	Implementation	Reference
CKG 1	Key Type:Asymmetric	N/A	SP 800-133rev2 Section 4/Example 1

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

The module implements the security functions listed in the table below.

Name	Type	Description	Properties	Algorithms
Symmetric Encryption/Decryption	BC-Auth BC-UnAuth			AES-CBC: (A4648) AES-CFB128: (A4648) AES-CTR: (A4648) AES-GCM: (A4648)
KAS ECC	KAS-SSC/KDF	Key agreement primitives	IG D.F:Scenario 2 path (2) Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	KAS-ECC Sp800-56Ar3: (A4648)
KAS FFC	KAS-SSC/KDF	Key agreement primitives	IG D.F:Scenario 2 path (2) Caveat:Key establishment methodology provides 112 bits of security strength	KAS-FFC Sp800-56Ar3: (A4648)

Name	Type	Description	Properties	Algorithms
KTS RSA	AsymKeyPair-Decap AsymKeyPair-Encap	Key encapsulation and un-encapsulation	IG D.G:Approved bullet 1 Caveat:Key establishment methodology provides 112 bits of security strength	KTS-IFC: (A4648)
KTS AES	BC-AuthDecrypt BC-AuthEncrypt	Authenticated encryption and decryption	IG D.G:Approved bullet 2 Caveat:Key establishment methodology provides 256 bits of security strength	AES-CFB128: (A4648) Key Length: 256 HMAC-SHA2-256: (A4648) SHA2-256: (A4648)
Message Authentication HMAC	MAC	Generation and verification of data integrity		HMAC-SHA-1: (A4648) HMAC-SHA2-256: (A4648) HMAC-SHA2-384: (A4648) HMAC-SHA2-512: (A4648) SHA-1: (A4648) SHA2-256: (A4648) SHA2-384: (A4648) SHA2-512: (A4648)
KBKDF	KBKDF	Generate AES Key-wrapping Key and HMAC keys		KDF SP800-108: (A4648) HMAC-SHA2-256: (A4648) SHA2-256: (A4648)
DRBG Request	DRBG	Request random data from DRBG		Hash DRBG: (A4648) SHA2-256: (A4648)
Entropy Source	ENT-Cond ENT-ESV	Request Entropy		SHA3-256: (A3449)
Signature Generation RSA	DigSig-SigGen			RSA SigGen (FIPS186-4): (A4648) SHA2-256: (A4648)
Signature Verification RSA	DigSig-SigVer			RSA SigVer (FIPS186-4): (A4648) SHA2-256: (A4648) SHA2-384: (A4648) SHA2-512: (A4648)
Signature Generation ECDSA	DigSig-SigGen			ECDSA SigGen (FIPS186-4): (A4648) SHA2-256: (A4648) SHA2-384: (A4648) SHA2-512: (A4648)
Signature Verification ECDSA	DigSig-SigVer			ECDSA SigVer (FIPS186-4): (A4648) SHA2-256: (A4648) SHA2-384: (A4648) SHA2-512: (A4648)
Generate Secure Hash	SHA			SHA-1: (A4648) SHA2-256: (A4648) SHA2-384: (A4648) SHA2-512: (A4648)

Name	Type	Description	Properties	Algorithms
Generate Asymmetric Keys 1 (RSA)	AsymKeyPair-KeyGen	Generate an asymmetric key pair	SP 800-133rev2 Sections 5.1:Asymmetric key generation using unmodified DRBG output	RSA KeyGen (FIPS186-4): (A4648) Hash DRBG: (A4648)
Generate Asymmetric Keys 2 (EC)	AsymKeyPair-KeyGen	Generate an asymmetric key pair	SP 800-133rev2 Sections 5.1 and 5.2:Asymmetric key generation using unmodified DRBG output	ECDSA KeyGen (FIPS186-4): (A4648) ECDSA KeyVer (FIPS186-4): (A4648) Hash DRBG: (A4648)
Generate Asymmetric Keys 3 (DH)	AsymKeyPair-KeyGen	Generate an asymmetric key pair	SP 800-133rev2 Sections 5.2:Asymmetric key generation using unmodified DRBG output	Safe Primes Key Generation: (A4648) Safe Primes Key Verification: (A4648) Hash DRBG: (A4648)

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

AES-GCM Key and IV generation for TLS

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.

2.8 RBG and 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, 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.

Cert Number	Vendor Name
E49	Senetas Corporation Ltd, distributed by Thales SA (SafeNet)

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Senetas CPU Jitter Entropy Source	Non-Physical	Intel® Xeon® D-2145NT (Skylake)	256 bits	Full Entropy	SHA3-256 (A3449)

Table 10: Entropy Sources

2.9 Key Generation

Asymmetric RSA and ECDSA keys are generated in accordance with FIPS186-4. Symmetric keys are generated using the SP 800-108rev1 compliant KDF function.

2.10 Key Establishment

The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer some or all of the underlying KAS cryptographic functionality to be used by an external operator/application as part of an approved KAS.

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

2.11 Industry Protocols

3 Cryptographic Module Interfaces

3.1 Ports and 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.

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API call variables. Data read from variables passed in the API.
N/A	Data Output	API return variables. Data written to user supplied variables or pointers in the API.
N/A	Control Input	API function calls. The API function called and the parameters by which it is invoked.
N/A	Status Output	API return values. The return value of the invoked API call.
N/A	Power	

Table 11: Ports and Interfaces

Note 1: The Control Output interface was intentionally omitted from this table, as the module does not implement it.

3.2 Data Privacy

During the pre-operational self-test phase the API is unavailable to the user and data cannot be input to or output from the module prior to successful completion of the self-tests.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

4.2 Roles

The cryptographic module supports a single role of Crypto Officer. The Crypto Officer has access to all approved services.

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	

Table 12: Roles

4.3 Approved Services

The module supports the approved services listed in the table below.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
AES Encryption/ Decryption		Status	API call parameters, AES Keys and cipher/plain text	Status, cipher/plain text	Symmetric Encryption/Decryption	Crypto Officer - AES Keys: W,E
RSA Key Generation		Status	API call parameters	Status, RSA Private and RSA Public Keys	Generate Asymmetric Keys 1 (RSA)	Crypto Officer - RSA Private Keys: G,R - RSA Public Keys: G,R
RSA Signature Generation and Verification		Status	API call parameters, RSA Private and RSA Public Keys, message	Status, Signature	Signature Generation RSA Signature Verification RSA	Crypto Officer - RSA Private Keys: W,E - RSA Public Keys: W,E
ECDSA Key Generation		Status	API call parameters	Status, ECDSA Private and ECDSA Public Keys	Generate Asymmetric Keys 2 (EC)	Crypto Officer - ECDSA Private Keys: G,R - ECDSA Public Keys: G,R
ECDSA Signature Generation and Verification		Status	API call parameters, ECDSA Private and ECDSA Public Keys, message	Status, Signature	Signature Generation ECDSA Signature Verification ECDSA	Crypto Officer - ECDSA Private Keys: W,E - ECDSA Public Keys: W,E
ECDH Key Agreement		Status	API call parameters	Status, Agreed Key	KAS ECC	Crypto Officer - ECDHE Private Keys: R,W,E - ECDHE Public Keys: R,W,E - ECDHE Shared Secret: G

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
DH Key Generation		Status	API call parameters	Status, DH Key Pair	Generate Asymmetric Keys 3 (DH)	Crypto Officer - Diffie Hellman Private Keys: G,R - Diffie Hellman Public Keys: G,R
DH Key Agreement		Status	API call parameters	Status, Agreed Key	KAS FFC	Crypto Officer - Diffie Hellman Private Keys: R,W,E - Diffie Hellman Public Keys: R,W,E - Diffie Hellman Shared Secret: G
Secure Hash Generation		Status	API call parameters, message	Status, Hash	Generate Secure Hash	Crypto Officer
HMAC Generation and Verification		Status	API call parameters, HMAC key, message	Status, HMAC	Message Authentication HMAC	Crypto Officer - HMAC Key: W,E
Random Number Generation		Status	API call parameters	Status, Random numbers	DRBG Request Entropy Source	Crypto Officer - DRBG Entropy Input and Nonce: G,E - DRBG Seed: G,E - DRBG V and C internal state parameters: G,E
Key Based Key Derivation Function		Status	API call parameters, KBKDF Key Derivation Key	Status, Derived Key	KBKDF	Crypto Officer - KBKDF Key Derivation Key: W,E - AES Key-wrapping Key: G,R - HMAC Key: G,R
RSA Key Encapsulation/ Un-encapsulation		Status	API call parameters, RSA Public and RSA Private Keys, AES key to be en/un-capsulated	Status, en/un-encapsulated AES key	KTS RSA	Crypto Officer - RSA Private Keys: W,E - RSA Public Keys: W,E
AES Key Wrapping/ Unwrapping		Status	API call parameters, AES Key-wrapping Key, AES Key to be un/wrapped	Status, un/wrapped AES key	KTS AES	Crypto Officer - AES Key-wrapping Key: W,E
Self-test	Run self-tests	N/A	API call parameters	Status (Test results: Pass/Fail)	None	Crypto Officer
Show Status	API call return code	N/A	API call parameters	Status	None	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show Module Info (Show Version)	API call (inventory), module version number and description	N/A	API call parameters	Status (Module version number and description)	None	Crypto Officer
Zeroisation	Zeroise all unprotected SSPs and key components within the module	N/A	API call parameters	N/A	None	Crypto Officer - AES Keys: Z - RSA Private Keys: Z - RSA Public Keys: Z - ECDSA Private Keys: Z - ECDSA Public Keys: Z - ECDHE Private Keys: Z - ECDHE Public Keys: Z - ECDHE Shared Secret: Z - Diffie Hellman Private Keys: Z - Diffie Hellman Public Keys: Z - Diffie Hellman Shared Secret: Z - DRBG Entropy Input and Nonce: Z - DRBG Seed: Z - DRBG V and C internal state parameters: Z - KBKDF Key Derivation Key: Z - AES Key-wrapping Key: Z - HMAC Key: Z

Table 13: Approved Services

The abbreviations of the access rights to SSPs have the following interpretation:

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.

4.4 Non-Approved Services

N/A for this module.

4.5 External Software/Firmware Loaded

The module does not support loading of external firmware.

5 Software/Firmware Security

5.1 Integrity Techniques

The approved SHA-256 algorithm implemented in the module is used to verify the integrity of all components within the cryptographic boundary of the module. If this integrity test fails, the module stays in the pre-operational state, preventing access to the API and all cryptographic services.

5.2 Initiate on Demand

The user can execute the Software/Firmware Integrity Test on demand by rebooting the module's host platform.

6 Operational Environment

6.1 Operational Environment Type and Requirements

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

Type of Operational Environment: Non-Modifiable

How Requirements are Satisfied:

As per ISO/IEC 19790:2012 7.6.3:

• The cryptographic module has control over its own SSPs.

• The host operating system provides process isolation and memory protection mechanisms that prevent uncontrolled access to the CSPs and uncontrolled modification of the SSPs. This ensures that direct access to SSPs is restricted to the cryptographic module and the trusted parts of the operational environment.

• Processes that are spawned by the cryptographic module are owned by the module and are not owned by external processes/operators.

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

Currently, the ISO/IEC 19790:2012 non-invasive security area is not required by FIPS 140-3 (see NIST SP 800-140F). The requirements of this area are not applicable to the module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

The module's SSP storage areas are detailed in the table below.

Storage Area Name	Description	Persistence Type
RAM	Temporary storage of SSPs in main dynamic memory.	Dynamic

Table 14: Storage Areas

9.2 SSP Input-Output Methods

The module's SSP input-output methods are detailed in the table below.

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API Call Input	Calling Application	RAM	Plaintext	Manual	Electronic	
API Call Output	RAM	Calling Application	Plaintext	Manual	Electronic	

Table 15: SSP Input-Output Methods

9.3 SSP Zeroization Methods

The module's SSP zeroisation methods are detailed in the table below.

Zeroization Method	Description	Rationale	Operator Initiation
Power Cycle	Zeroization will be initiated immediately upon a module power cycle.	All SSPs are stored in volatile memory (RAM) and will be destroyed by power cycling the module.	

Table 16: SSP Zeroization Methods

9.4 SSPs

The module's SSPs are detailed in the two tables below.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES Keys		128 bits, 256 bits - 128 bits, 256 bits	Symmetric Key - CSP			Symmetric Encryption/Decryption
RSA Private Keys		2048 bits - 112 bits	Asymmetric Key - CSP	Generate Asymmetric Keys 1 (RSA)		KTS RSA Signature Generation RSA
RSA Public Keys		2048 bits - 112 bits	Asymmetric Key - PSP	Generate Asymmetric Keys 1 (RSA)		KTS RSA Signature Verification RSA
ECDSA Private Keys		P-256, P-384, P-521 - 128 bits, 192 bits, 256 bits	Asymmetric Key - CSP	Generate Asymmetric Keys 2 (EC)		Signature Generation ECDSA

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ECDSA Public Keys		P-256, P-384, P-521 - 128 bits, 192 bits, 256 bits	Asymmetric Key - PSP	Generate Asymmetric Keys 2 (EC)		Signature Verification ECDSA
ECDHE Private Keys		P-256, P-384, P-521 - 128 bits, 192 bits, 256 bits	Asymmetric Key - CSP	Generate Asymmetric Keys 2 (EC)		KAS ECC
ECDHE Public Keys		P-256, P-384, P-521 - 128 bits, 192 bits, 256 bits	Asymmetric Key - PSP	Generate Asymmetric Keys 2 (EC)		KAS ECC
ECDHE Shared Secret		P-256, P-384, P-521 - 128 bits, 192 bits, 256 bits	Shared Secret - CSP		KAS ECC	
Diffie Hellman Private Keys		2048 bits - 112 bits	Asymmetric Key - CSP	Generate Asymmetric Keys 3 (DH)		KAS FFC
Diffie Hellman Public Keys		2048 bits - 112 bits	Asymmetric Key - PSP	Generate Asymmetric Keys 3 (DH)		KAS FFC
Diffie Hellman Shared Secret		2048 bits - 112 bits	Shared Secret - CSP		KAS FFC	
DRBG Entropy Input and Nonce	Used as input for SP 800-90Ar1 Hash_DRBG	384 bits -	Input to DRBG - CSP	Entropy Source		DRBG Request
DRBG Seed	Used as input for SP 800-90Ar1 Hash_DRBG	440 bits -	Input to DRBG - CSP	DRBG Request		DRBG Request
DRBG V and C internal state parameters	The V and C parameters store the internal state of the SP 800-90rev1 DRBG.	440 bits -	DRBG State Variables - CSP	DRBG Request		DRBG Request
KBKDF Key Derivation Key	The KBKDF Key Derivation Key is used to separately derive the AES Key-wrapping Key and the HMAC key using an SP 800-108 compliant KDF.	256 bits -	Key Derivation Key - CSP			KBKDF
AES Key-wrapping Key		256 bits - 256 bits	Symmetric Key - CSP	KBKDF		KTS AES
HMAC Key	The HMAC keys are used to protect the integrity of messages.	256 bits - 256 bits	Symmetric Key - CSP	KBKDF		Message Authentication HMAC

Table 17: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Keys	API Call Input	RAM:Plaintext	Zeroised after use	Power Cycle	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
RSA Private Keys	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	RSA Public Keys:Paired With
RSA Public Keys	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	RSA Private Keys:Paired With
ECDSA Private Keys	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	ECDSA Public Keys:Paired With
ECDSA Public Keys	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	ECDSA Private Keys:Paired With
ECDHE Private Keys	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	ECDHE Public Keys:Paired With
ECDHE Public Keys	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	ECDHE Private Keys:Paired With
ECDHE Shared Secret		RAM:Plaintext	Zeroised after use	Power Cycle	ECDHE Private Keys:Agreed Using ECDHE Public Keys:Agreed Using
Diffie Hellman Private Keys	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	Diffie Hellman Public Keys:Paired With
Diffie Hellman Public Keys	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	Diffie Hellman Private Keys:Paired With
Diffie Hellman Shared Secret		RAM:Plaintext	Zeroised after use	Power Cycle	Diffie Hellman Private Keys:Agreed Using Diffie Hellman Public Keys:Agreed Using
DRBG Entropy Input and Nonce		RAM:Plaintext	Zeroised after use	Power Cycle	
DRBG Seed		RAM:Plaintext	Zeroised after use	Power Cycle	DRBG Entropy Input and Nonce:Created From
DRBG V and C internal state parameters		RAM:Plaintext	Zeroised after use	Power Cycle	DRBG Seed:Associated With
KBKDF Key Derivation Key	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	
AES Key-wrapping Key	API Call Input	RAM:Plaintext	Zeroised after use	Power Cycle	KBKDF Key Derivation Key:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	API Call Output				
HMAC Key	API Call Input API Call Output	RAM:Plaintext	Zeroised after use	Power Cycle	KBKDF Key Derivation Key:Derived From

Table 18: SSP Table 2

9.5 Transitions

Please see the latest revision of SP 800-131 and CMVP Programmatic Transitions page for transitions that may affect this module.

10 Self-Tests

10.1 Pre-Operational Self-Tests

The module's pre-operational self-tests are detailed in the table below.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
SHA2-256 (A4648)		Approved Integrity Technique	SW/FW Integrity	Module becomes operational	A 32-byte SHA2-256 hash is used to verify the integrity of all components within the cryptographic software when the module is powered up and on demand by issuing the reboot command. The SHA256 algorithm is tested using a KAT prior to the Software/ Firmware integrity test running. Upon any file error the system will not transition from the pre-operational state to the operational state

Table 19: Pre-Operational Self-Tests

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 the table above), is tested using a Cryptographic Algorithm Self-Test (CAST) prior to the Pre-Operational Software/Firmware Integrity Test.

The module's conditional self-tests are detailed in the table below.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC (A4648)	128-bit, 256-bit	KAT	CAST	Module becomes operational	Separate Encrypt and Decrypt	Power Up
AES-CFB128 (A4648)	128-bit, 256-bit	KAT	CAST	Module becomes operational	Separate Encrypt and Decrypt	Power Up
AES-GCM (A4648)	128-bit, 256-bit	KAT	CAST	Module becomes operational	Separate Encrypt and Decrypt	Power Up
ECDSA SigGen (FIPS186-4) (A4648)	P-256, P-384, P-521	KAT	CAST	Module becomes operational	SigGen	Power Up
ECDSA SigVer (FIPS186-4) (A4648)	P-256, P-384, P-521	KAT	CAST	Module becomes operational	SigVer	Power Up
Hash DRBG (A4648)	256-bit	KAT	CAST	Module becomes operational	SP 800-90 A Section 11.3 (Instantiate, Reseed, Generate and Un-instantiate)	Power Up
HMAC-SHA-1 (A4648)	160-bit	KAT	CAST	Module becomes operational	MAC	Power Up
HMAC-SHA2-256 (A4648)	256-bit	KAT	CAST	Module becomes operational	MAC	Power Up
HMAC-SHA2-384 (A4648)	384-bit	KAT	CAST	Module becomes operational	MAC	Power Up
HMAC-SHA2-512 (A4648)	512-bit	KAT	CAST	Module becomes operational	MAC	Power Up
KAS-ECC Sp800-56Ar3 (A4648)	P-256, P-384, P-521	KAT	CAST	Module becomes operational	SP 800-56Arev3 Key Agreement	Power Up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KAS-FFC Sp800-56Ar3 (A4648)	2048-bit	KAT	CAST	Module becomes operational	SP 800-56Arev3 Key Agreement	Power Up
KDF SP800-108 (A4648)	256-bit	KAT	CAST	Module becomes operational	Key Derivation	Power Up
KTS-IFC (A4648)	2048-bit	KAT	CAST	Module becomes operational	Separate Encrypt and Decrypt	Power Up
RSA SigGen (FIPS186-4) (A4648)	2048-bit	KAT	CAST	Module becomes operational	Sign/ Verify	Power Up
RSA SigVer (FIPS186-4) (A4648)	2048-bit	KAT	CAST	Module becomes operational	Sign/ Verify	Power Up
SHA-1 (A4648)	160-bit	KAT	CAST	Module becomes operational	Secure Hash	Power Up
SHA2-256 (A4648)	256-bit	KAT	CAST	Module becomes operational	Secure Hash	Power Up
SHA2-384 (A4648)	384-bit	KAT	CAST	Module becomes operational	Secure Hash	Power Up
SHA2-512 (A4648)	512-bit	KAT	CAST	Module becomes operational	Secure Hash	Power Up
ECDSA KeyGen (FIPS186-4) (A4648)	P-256, P-384, P-521	PCT	PCT	API Return Code	Key Pair Generation	Key Pair Generation
RSA KeyGen (FIPS186-4) (A4648)	2048-bit	PCT	PCT	API Return Code	Key Pair Generation	Key Pair Generation
Safe Primes Key Generation (A4648)	2048-bit	PCT	PCT	API Return Code	Key Pair Generation	Key Pair Generation
Adaptive Proportion Test (APT)		FD	CAST	Module becomes operational	SP 800-90B Section 4	Power Up, Continuous, On Demand
Repeat Count Test (RCT)		FD	CAST	Module becomes operational	SP 800-90B Section 4	Power Up, Continuous, On Demand

Table 20: Conditional Self-Tests

10.3 Periodic Self-Test Information

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

The module's periodic self-tests are detailed in the table below.

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-256 (A4648)	Approved Integrity Technique	SW/FW Integrity	On Demand	Manually by power cycling module

Table 21: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A4648)	KAT	CAST	On Demand	Manually by power cycling module
AES-CFB128 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
AES-GCM (A4648)	KAT	CAST	On Demand	Manually by power cycling module
ECDSA SigGen (FIPS186-4) (A4648)	KAT	CAST	On Demand	Manually by power cycling module
ECDSA SigVer (FIPS186-4) (A4648)	KAT	CAST	On Demand	Manually by power cycling module
Hash DRBG (A4648)	KAT	CAST	On Demand	Manually by power cycling module
HMAC-SHA-1 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
HMAC-SHA2-256 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
HMAC-SHA2-384 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
HMAC-SHA2-512 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
KAS-ECC Sp800-56Ar3 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
KAS-FFC Sp800-56Ar3 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
KDF SP800-108 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
KTS-IFC (A4648)	KAT	CAST	On Demand	Manually by power cycling module
RSA SigGen (FIPS186-4) (A4648)	KAT	CAST	On Demand	Manually by power cycling module
RSA SigVer (FIPS186-4) (A4648)	KAT	CAST	On Demand	Manually by power cycling module
SHA-1 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
SHA2-256 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
SHA2-384 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
SHA2-512 (A4648)	KAT	CAST	On Demand	Manually by power cycling module
ECDSA KeyGen (FIPS186-4) (A4648)	PCT	PCT	On Demand	After key pair generation
RSA KeyGen (FIPS186-4) (A4648)	PCT	PCT	On Demand	After key pair generation

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Safe Primes Key Generation (A4648)	PCT	PCT	On Demand	After key pair generation
Adaptive Proportion Test (APT)	FD	CAST	On Demand	Manually by power cycling module
Repeat Count Test (RCT)	FD	CAST	On Demand	Manually by power cycling module

Table 22: Conditional Periodic Information

10.4 Error States

The module's self-test error states are described in the table below.

Name	Description	Conditions	Recovery Method	Indicator
Pre-operational	The module will remain in the pre-operational state until the pre-operational self-tests have successfully run. Cryptographic functions are inhibited while the module is in this state.	* Pre-Operational Software/ Firmware Integrity test failure * Cryptographic KAT failure * RCT/APT failure	Attempt to recover by rebooting the module.	pre-operational flag
Soft Error	Service operation aborted and error code returned. System state unchanged.	* PCT failure	Re-run service and observe return code	API return code

Table 23: Error States

10.5 Operator Initiation of Self-Tests

The Crypto Officer can run the pre-operational self-tests on demand by power cycling the host platform.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

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.

11.2 Administrator Guidance

Once powered on the module will run the pre-operational self-tests. On successful completion of the tests the module will automatically transition to the operational state (Approved mode) and the API will be available.

11.3 Non-Administrator Guidance

There is no non-Administrator guidance for the module.

11.4 End of Life

The module does not store SSPs persistently and can be sanitised by zeroisation.

12 Mitigation of Other Attacks

12.1 Attack List

The requirements in this section are Not Applicable.