Entrust Corporation

nShield 5s Hardware Security Module

FIPS 140-3 Non-Proprietary Security Policy

Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document defines the non-proprietary Security Policy enforced by the nShield 5s Hardware Security Module, i.e. the Cryptographic Module, to meet with the security requirements in FIPS 140-3 and ISO/IEC 19790.

The Cryptographic Module meets overall **FIPS 140-3 Security Level 3**.

1.2 Security Levels

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

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The nShield 5s Hardware Security Module (HSM) is a multi-chip embedded hardware Cryptographic Module as defined in FIPS 140-3, which comes in a PCI express board form factor protected by a tamper resistant enclosure, and performs encryption, digital signing, and key management on behalf of an extensive range of commercial and custom-built applications including public key infrastructures (PKIs), identity management systems, application-level encryption and tokenization, SSL/TLS, and code signing.

The nShield 5s HSM is also embedded inside the nShield 5c or the nShield HSMi, which are network attached appliances delivering cryptographic services as a shared network resource for distributed applications and virtual machines, giving organizations a highly secure solution for establishing physical and logical controls for server-based systems.

The Figures below show the nShield 5s HSM (left, representative of the two hardware variants nC5536E, nC5536N) and the nShield 5c appliance (right).

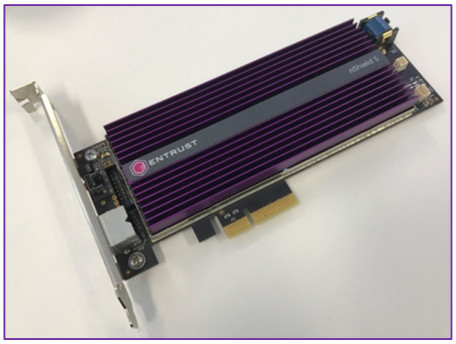


Figure 1 - nShield 5s

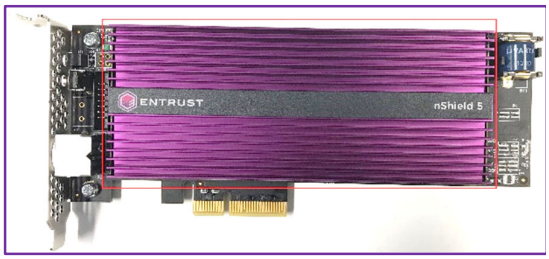
Figure 2 - nShield 5c

Module Type: Hardware

Module Embodiment: Multi-Chip Embedded

Cryptographic Boundary:

The cryptographic boundary is delimited in red in the images in the table below. It is delimited by the heat sink and the outer edge of the potting material on the top and bottom of the PCB.



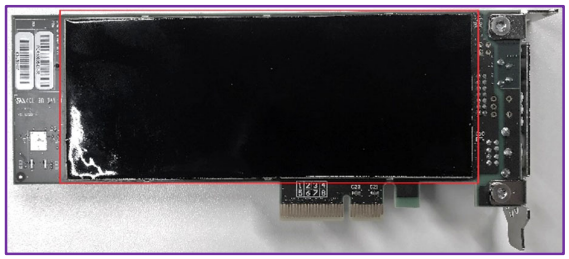


Figure 3 - Cryptographic boundary (Heat sink) **Figure 4 - Cryptographic boundary (Potting side)**

The module enforces that only approved services are available and plaintext import/export of secret or private keys is not allowed.

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
nShield 5s F3 (nC5536E)	PCB Assembly Part Number: PCA10005-01, PCB Assembly Revision: 03, 04	primary: 13.4.5, recovery: 13.2.4, uboot: 1.1.0, 1.4.1, 14.2.2	NXP QorIQ T1042 with cryptographic acceleration	PCIe form factor
nShield 5s for nShield 5c and for nShield HSMi (nC5536N)	PCB Assembly Part Number: PCA10005-01, PCB Assembly Revision: 03, 04	primary: 13.4.5, recovery: 13.2.4, uboot: 1.1.0, 1.4.1, 14.2.2	NXP QorIQ T1042 with cryptographic acceleration	PCIe form factor identical to the nShield 5s F3 (nC5536E), embedded inside the nShield 5c or the nShield HSMi network appliances

Table 2: Tested Module Identification – Hardware

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

N/A for this module.

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

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

N/A for this module.

2.3 Excluded Components

None.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
FIPS level 3 mode	In this mode the module is operating in approved mode. Only approved services are available.	Approved	Active modes: UseFIPSApprovedInternalMechanisms FIPSLevel3Enforcedv2 StrictSP80056Ar3

Table 3: Modes List and Description

Mode Change Instructions and Status:

To configure the cryptographic module in approved mode, the following steps are performed:

When the cryptographic module is in factory state, it first needs to be initialized with the Entrust supplied utility hsmadmin enroll.

Create a FIPS 140-3 level 3 compliant Security World using the Entrust supplied utility newworld, setting the mode to fips-140-level-3, e.g. `new-world --mode=fips-140-level-3`

An operator can verify that the module is configured in approved mode with the command line utility enquiry, which reports the following active modes:

```
active modes    UseFIPSApprovedInternalMechanisms
FIPSLevel3Enforcedv2 StrictSP80056Ar3
```

Once a FIPS 140-3 level 3 Security World is created, it is not possible to switch into a non-compliant mode without first zeroising the unprotected SSPs.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3707	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3707	Direction - Generation, Verification Key Length - 128, 192, 256 MAC Length - MAC Length: 128 Message Length - Message Length: 0-524288 Increment 8	SP 800-38B
AES-CTR	A3706	Direction - Decrypt, Encrypt Key Length - 128 Payload Length - Payload Length: 128	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
		Supports Counter larger than maximum value - No Incremental Counter - Yes Counter Tests Performed - Yes	
AES-ECB	A3706	Direction - Decrypt, Encrypt Key Length - 128	SP 800-38A
AES-ECB	A3707	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A3706	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128 Tag Length - 128 IV Length - IV Length: 96 Payload Length - Payload Length: 1024, 1024 AAD Length - AAD Length: 1024	SP 800-38D
AES-GCM	A3707	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.2 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 96 IV Length - IV Length: 96 Payload Length - Payload Length: 1024, 1024 AAD Length - AAD Length: 1024	SP 800-38D
AES-KW	A3707	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 128, 192, 256, 320, 4096	SP 800-38F
AES-KWP	A3707	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 128, 192, 256, 320, 4096	SP 800-38F
DSA KeyGen (FIPS186-4)	A3707	L - 2048, 3072 N - 224, 256	FIPS 186-4
DSA PQGGen (FIPS186-4)	A3707	P/Q Generation Methods - Probable G Generation Methods - Unverifiable L - 2048, 3072 N - 224, 256 Hash Algorithm - SHA2-224, SHA2-256	FIPS 186-4
DSA PQGVer (FIPS186-4)	A3707	P/Q Generation Methods - Probable G Generation Methods - Unverifiable L - 1024, 2048, 3072	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
		N - 160, 224, 256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256	
DSA SigGen (FIPS186-4)	A3707	L - 2048, 3072 N - 224, 256 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
DSA SigVer (FIPS186-4)	A3707	L - 1024, 2048, 3072 N - 160, 224, 256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA KeyGen (FIPS186-4)	A3707	Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3707	Curve - B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3706	Curve - P-256, P-521 Hash Algorithm - SHA2-256, SHA2-512	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3707	Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3706	Curve - P-256, P-521 Hash Algorithm - SHA2-256, SHA2-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3707	Curve - B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
Hash DRBG	A3707	Prediction Resistance - No Supports Reseed - Yes Mode - SHA2-256 Entropy Input - Entropy Input: 256 Nonce - Nonce: 128 Personalization String Length - Personalization String Length: 0-256 Increment 8 Additional Input - Additional Input: 0-256 Increment 8 Returned Bits - 1024	SP 800-90A Rev. 1
HMAC-SHA-1	A3707	MAC - MAC: 80-160 Increment 8 Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-224	A3707	MAC - MAC: 112-224 Increment 8 Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3706	MAC - MAC: 256 Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-256	A3707	MAC - MAC: 128-256 Increment 8 Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3707	MAC - MAC: 192-384 Increment 8 Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3707	MAC - MAC: 256-512 Increment 8 Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1
HMAC-SHA3-224	A3707	MAC - MAC: 112-224 Increment 8 Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1
HMAC-SHA3-256	A3707	MAC - MAC: 128-256 Increment 8 Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1
HMAC-SHA3-384	A3707	MAC - MAC: 192-384 Increment 8 Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1
HMAC-SHA3-512	A3707	MAC - MAC: 256-512 Increment 8 Key Length - Key Length: 128-2048 Increment 8	FIPS 198-1
KAS-ECC Sp800-56Ar3	A3707	Domain Parameter Generation Methods - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 Function - Full Validation, Key Pair Generation iutId - 0123456789CAFE Scheme - fullMqv - KAS Role - Initiator KDF Methods - oneStepKdf - Auxiliary Function Methods - Auxiliary Function Name - SHA2-224 Fixed Info Pattern - label||literal[00]||context||||uPartyInfo||vPartyInfo Fixed Info Encoding - Concatenation Key Length - 256	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A3706	Domain Parameter Generation Methods - P-256 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
KAS-FFC Sp800-56Ar3	A3707	Domain Parameter Generation Methods - FB, FC, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 Function - Full Validation, Key Pair Generation iutId - 0123456789CAFE Scheme - dhEphem - KAS Role - Initiator KDF Methods - oneStepKdf - Auxiliary Function Methods - Auxiliary Function Name - SHA2-224 MAC Salting Methods - default Fixed Info Pattern - label||literal[00]||context|||uPartyInfo||vPartyInfo Fixed Info Encoding - Concatenation Key Length - 256 dhOneFlow - KAS Role - Initiator, Responder KDF Methods - oneStepKdf - Auxiliary Function Methods - Auxiliary Function Name - SHA2-224 MAC Salting Methods - default Fixed Info Pattern - label||literal[00]||context|||uPartyInfo||vPartyInfo Fixed Info Encoding - Concatenation Key Length - 256 dhStatic - KAS Role - Initiator KDF Methods - oneStepKdf - Auxiliary Function Methods - Auxiliary Function Name - SHA2-224 MAC Salting Methods - default Fixed Info Pattern - label||literal[00]||context|||uPartyInfo||vPartyInfo Fixed Info Encoding - Concatenation Key Length - 256	SP 800-56A Rev. 3
KDF SP800-108	A3707	KDF Mode - Counter MAC Mode - CMAC-AES256 Supported Lengths - Supported Lengths: 128, 192, 256 Fixed Data Order - Before Fixed Data Counter Length - 8 Supports Empty IV - No	SP 800-108 Rev. 1

Algorithm	CAVP Cert	Properties	Reference
		Requires Empty IV - No Custom Key In Length - 0	
KDF SSH (CVL)	A3706	Cipher - AES-128 Hash Algorithm - SHA2-256	SP 800-135 Rev. 1
KMAC-128	A3707	Message Length - Message Length: 0-65536 Increment 8 MAC Length - MAC Length: 64-32768 Increment 8 Key Data Length - Key Data Length: 128-2048 Increment 8 Hex Customization - No Supports eXtendable-Output Functions - No	SP 800-185
KMAC-256	A3707	Message Length - Message Length: 0-65536 Increment 8 MAC Length - MAC Length: 64-32768 Increment 8 Key Data Length - Key Data Length: 256-2048 Increment 8 Hex Customization - No Supports eXtendable-Output Functions - No	SP 800-185
KTS-IFC	A3707	Function - partialVal IUT ID - CAFECAFE Modulo - 2048, 3072, 4096 Key Generation Methods - rsakpg1-basic Fixed Public Exponent - 010001 Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Hash Algorithms - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Supports Null Associated Data - Yes Associated Data Encoding - concatenation Key Length - 512	SP 800-56B Rev. 2
RSA KeyGen (FIPS186-4)	A3707	Key Generation Mode - B.3.6 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Info Generated By Server - No Public Exponent Mode - Fixed Fixed Public Exponent - 010001 Private Key Format - Chinese Remainder Theorem	FIPS 186-4
RSA SigGen (FIPS186-4)	A3707	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA2-224	FIPS 186-4
RSA SigVer (FIPS186-4)	A2404	Signature Type - PKCS 1.5 Modulo - 4096 Hash Pair -	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
		Hash Algorithm - SHA2-256 Public Exponent Mode - Random	
RSA SigVer (FIPS186-4)	A3707	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA-1 Public Exponent Mode - Random	FIPS 186-4
RSA SigVer (FIPS186-4)	A6385	Signature Type - PKCS 1.5 Modulo - 4096 Hash Pair - Hash Algorithm - SHA2-256 Public Exponent Mode - Random	FIPS 186-4
Safe Primes Key Generation	A3707	Safe Prime Groups - MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3
Safe Primes Key Verification	A3707	Safe Prime Groups - MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3
SHA-1	A3707	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-224	A3707	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-256	A2404	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-256	A3706	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-256	A3707	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-256	A6385	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-384	A3707	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512	A3706	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512	A3707	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA3-224	A3707	Message Length - Message Length: 8-51200 Increment 8	FIPS 202
SHA3-256	A3707	Message Length - Message Length: 8-51200 Increment 8	FIPS 202
SHA3-384	A3707	Message Length - Message Length: 8-51200 Increment 8	FIPS 202
SHA3-512	A3707	Message Length - Message Length: 8-51200 Increment 8	FIPS 202

Table 4: Approved Algorithms

The following notes and caveats apply to the nCore crypto library:

- For AES GCM, the 96-bit IV is internally generated using the approved DRBG as per IG C.H.

- ECDSA SigVer with P-192 has been CAVP-tested but is 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.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key type:Symmetric	n/a	SP800-133r2 section 4/example1 and section 6.1

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

Name	Properties	Implementation	Reference
ECDSA	Brainpool curves:brainpoolP224r1/P224t1 (112 bits of strength), brainpoolP256r1/P256t1 (128 bits of strength), brainpoolP320r1/P320t1 (160 bits of strength), brainpoolP384r1/P384t1 (192 bits of strength), brainpoolP512r1/P512t1 (256 bits of strength)	nShield 5 Algorithm Library - nCore	SP800-186
KAS-ECC	Brainpool curves:brainpoolP224r1/P224t1 (112 bits of strength), brainpoolP256r1/P256t1 (128 bits of strength), brainpoolP320r1/P320t1 (160 bits of strength), brainpoolP384r1/P384t1 (192 bits of strength), brainpoolP512r1/P512t1 (256 bits of strength)	nShield 5 Algorithm Library - nCore	SP800-186

Table 6: Non-Approved, Allowed Algorithms

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

Only approved and non-approved but allowed cryptographic algorithms are supported.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Platform RBG	ENT-P DRBG CKG	RBG2 construction with a physical entropy source	Strength:256	Hash DRBG: (A3707) CKG: (n/a) Key type: Symmetric

Name	Type	Description	Properties	Algorithms
SSH lib: Symmetric Ciphers	BC-Auth BC-UnAuth	Symmetric Encrypt/Decrypt	key size:128 bits	AES-GCM: (A3706) AES-CTR: (A3706) AES-ECB: (A3706)
SSH lib: Digital Signature	DigSig-SigGen DigSig-SigVer	ECDSA SigGen / SigVer	Strength:128, 256 bits	ECDSA SigGen (FIPS186-4): (A3706) ECDSA SigVer (FIPS186-4): (A3706)
SSH lib: Message Authentication	MAC	Generate or verify data integrity	Strength:256 bits	HMAC-SHA2-256: (A3706)
SSH lib: Key Agreement	KAS-SSC KAS-135KDF	Key agreement	Strength:128 bits	KAS-ECC-SSC Sp800-56Ar3: (A3706) KDF SSH: (A3706)
SSH lib: Message digest	SHA	Hashing - ssh	Strength:128, 256 bits	SHA2-256: (A3706) SHA2-512: (A3706)
Bootloader: Digital signature verification	DigSig-SigVer	Signature verification performed by the bootloader	Strength:128 bits	RSA SigVer (FIPS186-4): (A2404, A6385) SHA2-256: (A2404, A6385)
nCoreAPI RBG	DRBG CKG	Root RBGC construction with Platform RBG as the randomness source	Strength:256	Hash DRBG: (A3707) CKG: (n/a) Key type: Symmetric
nCore lib: FW integrity	DigSig-SigVer	Signature verification performed by the bootloader	Strength:256 bits	ECDSA SigVer (FIPS186-4): (A3707)
nCore lib: Key Blobs	BC-UnAuth MAC KBKDF KTS-Encap	nShield Key blob	key size:256 bits	AES-CBC: (A3707) HMAC-SHA2-256: (A3707) KDF SP800-108: (A3707) KTS-IFC: (A3707)

Name	Type	Description	Properties	Algorithms
nCore lib: Smartcard shares	BC-UnAuth MAC KBKDF	Encrypted shares stored in smartcards	key size:256 bits	AES-CBC: (A3707) HMAC-SHA2-256: (A3707) KDF SP800-108: (A3707)
nCore lib: Impath session key establishment	KAS-Full	Key agreement of an Impath session	IG:IG D.F Scenario 2, path (2), end to end Caveat:Key establishment methodology provides 128 bits of security strength	KAS-FFC Sp800-56Ar3: (A3707) Safe Primes Key Generation: (A3707) Safe Primes Key Verification: (A3707)
nCore lib: Impath messaging	BC-Auth BC-UnAuth MAC	Impath session message confidentiality and integrity	Strength:128 bits	AES-GCM: (A3707) AES-CBC: (A3707) HMAC-SHA2-256: (A3707)
nCore lib: Symmetric Ciphers	BC-Auth BC-UnAuth	Symmetric ciphers offered by the nCore API	Strength:112 to 256 bits	AES-GCM: (A3707) AES-CBC: (A3707) AES-ECB: (A3707) AES-CTR: (A3707)
nCore lib: Key wrapping	KTS-Wrap	Symmetric key wrapping offered by the nCore API	Strength:128, 192, 256 bits	AES-GCM: (A3707) AES-KW: (A3707) AES-KWP: (A3707)
nCore lib: Digital signature	DigSig-SigGen DigSig-SigVer	Digital signature offered by the nCore API	Strength:112 to 256 bits	DSA SigGen (FIPS186-4): (A3707) DSA SigVer (FIPS186-4): (A3707) ECDSA SigGen (FIPS186-4): (A3707) ECDSA SigVer (FIPS186-4): (A3707) RSA SigGen (FIPS186-4): (A3707)

Name	Type	Description	Properties	Algorithms
				RSA SigVer (FIPS186-4): (A3707)
nCore lib: Asymmetric key generation	AsymKeyPair-KeyGen AsymKeyPair-KeyVer AsymKeyPair-DomPar	Asymmetric key generation offered by the nCore API	Strength:112 to 256 bits	ECDSA KeyGen (FIPS186-4): (A3707) ECDSA KeyVer (FIPS186-4): (A3707) RSA KeyGen (FIPS186-4): (A3707) DSA KeyGen (FIPS186-4): (A3707) DSA PQGGen (FIPS186-4): (A3707) DSA PQGVer (FIPS186-4): (A3707) Safe Primes Key Generation: (A3707) Safe Primes Key Verification: (A3707)
nCore lib: Message authentication	MAC	MAC algorithms offered by the nCore API	Strength:112 to 256 bits	HMAC-SHA-1: (A3707) HMAC-SHA2-224: (A3707) HMAC-SHA2-256: (A3707) HMAC-SHA2-384: (A3707) HMAC-SHA2-512: (A3707) HMAC-SHA3-224: (A3707) HMAC-SHA3-256: (A3707) HMAC-SHA3-384: (A3707) HMAC-SHA3-512: (A3707)

Name	Type	Description	Properties	Algorithms
				KMAC-128: (A3707) KMAC-256: (A3707) AES-CMAC: (A3707)
nCore lib: Secure hashing	SHA	Hash algorithms offered by the nCore API	Strength:112 to 256 bits	SHA-1: (A3707) SHA2-224: (A3707) SHA2-256: (A3707) SHA2-384: (A3707) SHA2-512: (A3707) SHA3-224: (A3707) SHA3-256: (A3707) SHA3-384: (A3707) SHA3-512: (A3707)
nCore lib: Key establishment	KAS-Full	Key agreement functions offered by the nCore API	IG:IG D.F Scenario 2, path (2), end to end Caveat:Key establishment methodology provides between 112 and 256 bits of security strength	KAS-ECC Sp800-56Ar3: (A3707) KAS-FFC Sp800-56Ar3: (A3707)
nCore lib: Key transport	KTS-Encap	Key transport functions offered by the nCore API	Strength:112 to 128 bits	KTS-IFC: (A3707)
nCore lib: KDFs	KBKDF	Key derivation functions offered by the nCore API	Strength:128, 192, 256 bits	KDF SP800-108: (A3707)
nCore lib: Remote Admin session key establishment	KAS-Full	Key agreement of a Remote Admin session	IG:IG D.F Scenario 2, path (2), end to end Caveat:Key establishment	KAS-ECC Sp800-56Ar3: (A3707)

Name	Type	Description	Properties	Algorithms
			methodology provides 256 bits of security strength	
nCore lib: Remote Admin messaging	BC-UnAuth MAC	Remote Admin session message confidentiality and integrity	Strength:256 bits	AES-CBC: (A3707) AES-CMAC: (A3707)

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

n/a

2.8 RBG and Entropy

Cert Number	Vendor Name
E38	Entrust

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
nShield 5s Physical True Random Number Generator	Physical	NXP® QorIQ T1042 with cryptographic acceleration	1	0.89 bits	none

Table 9: Entropy Sources

The cryptographic module's physical entropy source is a hardware based true random number generator used to seed the DRBGs.

2.9 Key Generation

The cryptographic module generates symmetric keys according to SP 800-133 section 6.1 "Direct Generation".

2.10Key Establishment

The cryptographic module implements the key establishment methods specified in Security Function Implementations.

2.11 Industry Protocols

As per IG D.C., no parts of the SSH protocol, other than the approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP.

For AES GCM in the SSH crypto library, the module is compliant with RFCs 4252, 4253 and 5647, and the IV is generated according to the SSHv2 protocol IV generation, as per IG C.H. In case the module's power is lost and then restored, a new key for use with the AES-GCM encryption/decryption is established.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
PCIe bus	Data Input Data Output Control Input Status Output Power	nCoreAPI, SSHAdmin, updater, setup, monitor, launcher, discovery, reboot, start connection
Smartcard reader serial port	Data Output	APDU commands
Status LED	Control Input	n/a
Recovery button	Status Output	n/a
Battery	Power	n/a

Table 10: Ports and Interfaces

The following figure shows the module's physical ports:

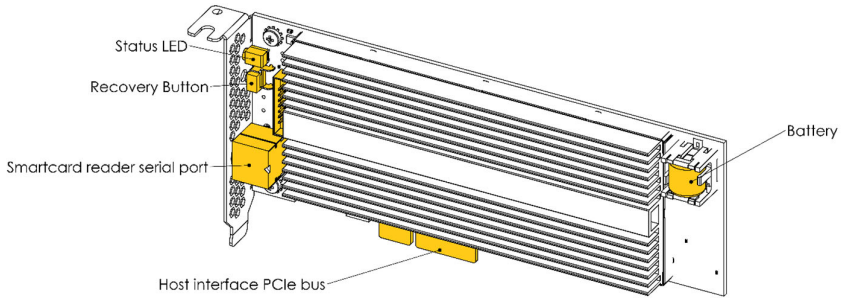


Figure 5 - Physical ports

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
SSH authentication	ECDSA client key identity-based authentication	SSH lib: Digital Signature	This results in a security strength of 128 bits. A random authentication attempt gives is a probability of success of 2^ -128, which is less than one in 1,000,000.	The module can process around 2^20 commands per minute. This gives a probability of success in a one minute period of 2^-108, which is less than one in 100,000.
Smartcard authentication	Smartcard identity-based authentication	nCore lib: Smartcard shares	A logical token share stored in a Smartcard or Softcard is encrypted and MAC'ed. An attacker would need to guess the encrypted share value and the associated MAC in order to be able to load a valid Logical token share into the module. This requires, as a minimum, guessing a 256-bit HMAC-SHA256 value, which gives a security strength of 256 bits. A random authentication attempt gives is a probability of success of 2^ -256 , which is less than one in 1,000,000.	The module can process around 2^20 commands per minute. This gives a probability of success in a one minute period of 2^-236, which is less than 10^-5 , which is less than one in 100,000.

Table 11: Authentication Methods

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Platform Crypto Officer (PCO)	Identity	Crypto Officer	SSH authentication

Name	Type	Operator Type	Authentication Methods
nShield Security Officer (NSO)	Identity	Crypto Officer	Smartcard authentication
User Client (UC)	Identity	User	SSH authentication

Table 12: Roles

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
setup info (Show Version)	Show Module's Versioning Information service. This command prints out the contents of the board-id rom file and a number of flags that indicate which other setup subcommands have been previously executed as determined by the existence or non-existence of the relevant files in long-term storage. It also prints out the tag and value pairs of any	return status code 0	input arguments	versioning information, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	options set with the setopt subcommand.					
setup factorystate	Performs zeroisation of unprotected SSPs and returns the module to factory state.	return status code 0	input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication Platform RBG	Platform Crypto Officer (PCO) - KRESET: Z,G - KSESSION_SSH: E - KSSH_SETUP: G - KSSH_SSHADMIN: G - KSSH_NCORE: G - KSSH_MONITOR: G - KSSH_UPDATER: G - KSSH_LAUNCHER: G - DRBG-1 entropy input: E - DRBG-1 seed, Internal State: C and V values: E
setup gettime	Returns the system date and time.	return status code 0	input arguments	time, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
setup settime	Sets the system date and time.	return status code 0	input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E
sshadmin set	Loads the client public key in the module, that will be used to authenticate the requester of a particular service	return status code 0	Input arguments, service, SSH client public key	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E - KSSH_CLIENT pub: W
sshadmin list	Obtains the client public key for the service given	return status code 0	Input arguments, service	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E - KSSH_CLIENT pub: R
sshadmin get-serverkey	Obtains the server public key for the service given	return status code 0	Input arguments, service	SSH server public key, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E - KSSH_NCORE pub: R - KSSH_UPDATER pub: R - KSSH_SETUP pub: R - KSSH_LAUNCHER pub: R - KSSH_SSHAD

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						MIN pub: R - KSSH_MONITOR pub: R
updater info (Show Version)	This is a Show Module's Versioning Information service. Obtains the version number of the HSM firmware	return status code 0	input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH : E
updater receive	Transmits a file (intended to be an npkg upgrade file) to the HSM.	return status code 0	input arguments, fw update file	fw update file info, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH : E
updater load	Verifies that a file on the HSM filesystem is a valid npkg upgrade file and, if so loads the file onto its flash partition	return status code 0	Input arguments, fw update file info	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH : E - NPSK pub: E,W
updater setminvsn	Sets the minimum VSN	return status code 0	input arguments, vsn	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH : E
monitor getlog	Obtains the log of the system	return status code 0	input arguments	logs, return	SSH lib: Symmetric Ciphers SSH lib:	Platform Crypto Officer (PCO) -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
				status code	Message Authentication	KSESSION_SSH: E
monitor clearlog	Clears the log of the system	return status code 0	input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E
monitor getenvstats	Obtains the environmental statistics	return status code 0	input arguments	env data, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E
start connection	Initiates and establishes an ssh connection to a service	return status code 0	input arguments	return status code	SSH lib: Digital Signature SSH lib: Key Agreement SSH lib: Message digest Platform RBG	Platform Crypto Officer (PCO) - KUSER_SSH: E - KSSH_SETUP: E - KSSH_UPDATER: E - KSSH_SSHADMIN: E - KSSH_LAUNCHER: E - KSSH_MONITOR: E - KSSH_CLIENTpub: E User Client (UC) - KCONTAINERSSH: E - KSSH_NCORE:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						E - KSSH_CLIENT pub: E
discovery python-zeroconf	IP address discovery service	return status code 0	input arguments	IP address, return status code		Unauthenticated
reboot (Self test)	Reboots the HSM and runs the self-tests	n/a	n/a	n/a	Bootloader: Digital signature verification nCore lib: FW integrity	Unauthenticated - KRESET: E - KUSER_SSH: G,E - KCONTAINER: G,E - KCONTAINERSSH: G,E
launcher receive	Uploads a file to the launcher service for temporary storage.	return status code 0	Input arguments, file	file id, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E
launcher create machine	Creates a SEE machine container from a received file, after successfully being validated	return status code 0	Input arguments	uuid, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Digital signature	Platform Crypto Officer (PCO) - KSESSION_SSH: E - DSK pub: E
launcher list	Lists SEE machines along with their current states.	return status code 0	Input arguments	machine list, return status code	SSH lib: Symmetric Ciphers SSH lib: Message	Platform Crypto Officer (PCO) - KSESSION_SSH: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Authentication	
launcher start	Starts a SEE machine after successfully being validated.	return status code 0	Input arguments	ip, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Digital signature	Platform Crypto Officer (PCO) - KSESSION_SSH: E - DSK pub: E
launcher stop	Stops a SEE machine.	return status code 0	Input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E
launcher destroy	Deletes a SEE machine.	return status code 0	Input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	Platform Crypto Officer (PCO) - KSESSION_SSH: E
launcher ids commands	Management of the SEE machine signing certificates.	return status code 0	Input arguments	output arguments, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Digital signature	Platform Crypto Officer (PCO) - KSESSION_SSH: E - DSK pub: R,W,E - ESK pub: E
launcher see-log commands	Management of the SEE machine logs.	return status code 0	Input arguments	output arguments, return status code	SSH lib: Symmetric Ciphers SSH lib: Message	Platform Crypto Officer (PCO) - KSESSION_SSH: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Authentication	
nCoreAPI Big number operation	Performs an operation on a large integer	return status OK	input arguments	operation result, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Make blob	Creates a Key blob containing the key	return status OK	input arguments, key handle	key blob, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Key Blobs	User Client (UC) - KSESSION_SSH: E - BLOBKE: R - BLOBKM: R - KA: R - KRE_BLOBKEY: R - KR: R - KM: R - KNSO: R - LTX: R
nCoreAPI Bulk channel	Provides a bulk processing channel for crypto operations	return status OK	input arguments, data	operation result, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Symmetric Ciphers nCore lib: Message authentication nCore lib: Digital signature	User Client (UC) - KSESSION_SSH: E - KA: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
nCoreAPI Check user action	Determines whether the ACL associated with a key allows a specific operator defined action	return status OK	input arguments, key handle	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KNSO: R - KA: R
nCoreAPI Clear unit	Clears all keys, tokens and shares in RAM and causes the module to reboot	return status OK	input arguments, module mode	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KA: Z - KR: Z - IMPATHKE: Z - IMPATHKM: Z - RAKME: Z - RAKMA: Z
nCoreAPI Set module key	Allows a key to be stored internally as a Module key (KM) value	return status OK	input arguments, key handle	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Secure hashing	nShield Security Officer (NSO) - KSESSION_SSH: E - KM: W
nCoreAPI Remove module key	Deletes the KM with a given KM hash value from non-volatile memory	return status OK	input arguments, key hash	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	nShield Security Officer (NSO) - KSESSION_SSH: E - KM: Z
nCoreAPI Duplicate key handle	Creates a second instance of a key with the same ACL	return status OK	input arguments, key handle	key handle, return status code	SSH lib: Symmetric Ciphers SSH lib: Message	User Client (UC) - KSESSION_SSH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Authentication	: E - KA: R
nCoreAPI Enable feature	Feature enabling service	return status OK	input arguments, features	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH : E
nCoreAPI Encryption	Data encryption	return status OK	input arguments, mechanism, key handle, plaintext data, iv	encrypted data, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Symmetric Ciphers	User Client (UC) - KSESSION_SSH : E - KA: E
nCoreAPI Decryption	Data decryption	return status OK	input arguments, mechanism, key handle, encrypted data	decrypted data, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Symmetric Ciphers	User Client (UC) - KSESSION_SSH : E - KA: E
nCoreAPI Erase from smartcard /softcard	Removes a file or a share from a smartcard or softcard	return status OK	input arguments, slot, file info	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH : E
nCoreAPI Format token	Formats a smartcard or a softcard	return status OK	input arguments, slot	return status code	SSH lib: Symmetric Ciphers SSH lib: Message	User Client (UC) - KSESSION_SSH : E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Authentication	
nCoreAPI File operations	Performs file operations in the module	return status OK	input arguments, file info, operation	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH : E
nCoreAPI Force module to fail	Causes the module to enter a failure state	return status OK	input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH : E
nCoreAPI Generate prime number	Generates a random prime	return status OK	input arguments, length	Bignumber, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCoreAPI RBG	User Client (UC) - KSESSION_SSH : E - DRBG-2 entropy input: E - DRBG-2 seed, Internal State: C and V values: E - DRBG-1 entropy input: E - DRBG-1 seed, Internal State: C and V values: E
nCoreAPI Random number	Generates a random number from the Approved RBG	return status OK	input arguments, length	random bytes, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH : E - DRBG-2 entropy input: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					nCoreAPI RBG	- DRBG-2 seed, Internal State: C and V values: E - DRBG-1 entropy input: E - DRBG-1 seed, Internal State: C and V values: E
nCoreAPI Get ACL	Get the ACL of a given key	return status OK	input arguments, key handle	acl, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KA: R
nCoreAPI Get application data	Get the application data field from a key	return status OK	input arguments, key handle	application data, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KA: R
nCoreAPI Get challenge	Get a random challenge that can be used in fresh certificates	return status OK	input arguments	nonce, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCoreAPI RBG	User Client (UC) - KSESSION_SSH: E - DRBG-2 entropy input: E - DRBG-2 seed, Internal State: C and V values: E - DRBG-1 entropy input: E - DRBG-1 seed, Internal State:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						C and V values: E
nCoreAPI Get KLF2	Get a handle to the Module Long Term (KLF2) public key	return status OK	input arguments	key handle, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Get key information	Get the type, length and hash of a key	return status OK	input arguments, key handle	key info, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Secure hashing	User Client (UC) - KSESSION_SSH: E - KA: R
nCoreAPI Get module signing key	Get a handle to the KML public key	return status OK	input arguments	key handle, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KML: R
nCoreAPI Get list of slot in the module	Get the list of slots that are available from the module	return status OK	input arguments	slot info, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Get logical token info	Get information about a Logical Token: hash, state and number of shares	return status OK	input arguments, key handle	logical token info, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib:	User Client (UC) - KSESSION_SSH: E - LTX: R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Secure hashing	
nCoreAPI Get list of module keys	Get the list of the hashes of all module keys and the KNSO	return status OK	input arguments	hash of KNSO (HKNSO), hash of module keys, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Secure hashing	User Client (UC) - KSESSION_SSH: E - KM: R - KNSO: R - HKNSO: R
nCoreAPI Get module state	Returns unsigned data about the current state of the module	return status OK	input arguments	module attributes, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Get real time clock	Get the current time from the module Real Time Clock	return status OK	input arguments	time, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Get Share access control list	Get the Share's ACL	return status OK	input arguments, slot	acl, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - SHAREKE: E - SHAREKM: E - KM: E
nCoreAPI Get slot information	Get information about shares and files on a Smartcard that has been	return status OK	input arguments, slot	smartcard info, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	inserted in a module slot					
nCoreAPI Get ticket	Get a ticket (an invariant identifier) for a key	return status OK	input arguments	ticket, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Initialize unit	Causes the nCore API service in the pre-initialization state to enter the initialization state. When the module enters the initialization state, it erases all Module keys (KM), the module's signing key (KML), and the hash of the Security Officer's keys, HKNSO. It then generates a new KML and KM	return status OK	input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Secure hashing nCore lib: Asymmetric key generation nCoreAPI RBG	User Client (UC) - KSESSION_SSH: E - KA: Z,G - KRE_BLOBKEY: Z,G - KR: Z,G - KM: Z,G - KAL: Z,G - KML: Z,G - KNSO: Z,G - HKNSO: Z,G - LTX: Z,G - DRBG-2 entropy input: E - DRBG-2 seed, Internal State: C and V values: E - DRBG-1 entropy input: E - DRBG-1 seed, Internal State: C and V values: E
nCoreAPI Insert Softcard	Allocates memory on the module that is used to store the	return status OK	input arguments, slot	return status code	SSH lib: Symmetric Ciphers SSH lib: Message	User Client (UC) -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	logical token share and other data objects.				Authentication	KSESSION_SSH: E
nCoreAPI Remove Softcard	Removes a Softcard from the module. It returns the updated shares and deletes them from the module's memory.	return status OK	input arguments, slot	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Impath channel	Support for Impath channel. Requires Feature Enabled	return status OK	input arguments, data	data, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Impath session key establishment nCore lib: Impath messaging	User Client (UC) - KSESSION_SSH: E - IMPATHKE: G,E - IMPATHKM: G,E
nCoreAPI Key generation	Generates a cryptographic key of a given type with a specified ACL. It returns a handle to the key. Optionally, it returns a KML signed	return status OK	input arguments, key params, acl, app data	key handle, key generation cert, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Asymmetric key generation nCoreAPI RBG	User Client (UC) - KSESSION_SSH: E - KA: G - KML: E - DRBG-2 entropy input: E - DRBG-2 seed, Internal State: C and V values:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	certificate with the hash of the key and its ACL information				nCore lib: Digital signature	E - DRBG-1 entropy input: E - DRBG-1 seed, Internal State: C and V values: E
nCoreAPI Key import	Loads a plain text key into the module. If the module is initialized in approved mode, this service is available for public keys only.	return status OK	input arguments, wrapped key, acl, app data	key handle, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KA: W
nCoreAPI Derive key	Performs key wrapping, unwrapping, transport, exchange and derivation	return status OK	input arguments, mechanism, key handles	key handle, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Key wrapping nCore lib: Key establishment nCore lib: Key transport nCore lib: KDFs	User Client (UC) - KSESSION_SSH: E - KA: R,W
nCoreAPI Load blob	Load a Key blob into the module. It returns a handle to	return status OK	input arguments, blob data	key handle, return	SSH lib: Symmetric Ciphers SSH lib: Message	User Client (UC) - KSESSION_SSH: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	the key suitable for use with module services			status code	Authentication nCore lib: Key Blobs	- KA: W - KRE_BLOBKEY: W - KR: W - KM: W - KNSO: W - BLOBKE: E - BLOBKM: E
nCoreAPI Load logical token	Initiates loading a Logical Token from Shares, which can be loaded with the Read Share command.	return status OK	input arguments, logical token hash	key handle, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Generate logical token	Creates a new Logical Token with given properties and secret sharing parameters	return status OK	input arguments, hash of km	logical token hash, key handle, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCoreAPI RBG	User Client (UC) - KSESSION_SSH: E - KM: E - LTX: G
nCoreAPI Message digest	Computes the cryptographic hash of a given message	return status OK	input arguments, mechanism, data to be hashed	hashed data, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Secure hashing	User Client (UC) - KSESSION_SSH: E
nCoreAPI Modular exponentiation	Performs modular exponentiation on	return status OK	input arguments	operation result, return status code	SSH lib: Symmetric Ciphers SSH lib: Message	User Client (UC) - KSESSION_SSH: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	supplied values				Authentication	
nCoreAPI Module hardware information	Reports detailed hardware information	return status OK	input arguments	hw info, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI No operation	No operation	return status OK	input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Change Share passphrase	Updates the passphrase of a Share	return status OK	input arguments,slot, old pin, new pin	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Key Blobs	User Client (UC) - KSESSION_SSH: E - SHAREKE: G,E,R - SHAREKM: G,E,R - KM: E
nCoreAPI NVRAM allocate	Allocation in NVRAM	return status OK	input arguments, file info, acl	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	nShield Security Officer (NSO) - KSESSION_SSH: E
nCoreAPI NVRAM free	Deallocation from NVRAM	return status OK	input arguments, filename	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
nCoreAPI Operation on NVM list	Returns a list of files in NVRAM	return status OK	input arguments	files info, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Operation on NVM files	Operation on an NVRAM file	return status OK	input arguments, filename, operation	operation result, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Key export	Exports a key in plain text. In approved mode, only public keys can be exported	return status OK	input arguments, key handle	wrapped key, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KA: R
nCoreAPI Read file	Reads data from a file on a Smartcard or Softcard.	return status OK	input arguments, file info, slot	file data, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Read share	Reads a share from a Smartcard or Softcard. Once a quorum of shares have been loaded, the module reassembles the Logical Token.	return status OK	input arguments, slot, key handle	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Smartcard shares	User Client (UC) - KSESSION_SSH: E - SHAREKE: G,E,R - SHAREKM: G,E,R - LTX: G - KM: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
nCoreAPI Send Share to remote slot	Reads a Share and encrypts it with the Impath session keys for transmission to the peer module	return status OK	input arguments	encrypted share, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Smartcard shares nCore lib: Impath messaging	User Client (UC) - KSESSION_SSH: E - SHAREKE: G,E,R - SHAREKM: G,E,R - IMPATHKE: E - IMPATHKM: E
nCoreAPI Receive Share from remote slot		return status OK	input arguments	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Key Blobs nCore lib: Impath messaging	User Client (UC) - KSESSION_SSH: E - SHAREKE: W - SHAREKM: W - IMPATHKE: E - IMPATHKM: E
nCoreAPI Redeem ticket	Gets a handle in the current name space for the object referred to by a ticket created by Get Ticket.	return status OK	input arguments, ticket	key handle, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Remote administration	Provides remote presentation of Smartcards using a secure channel between the	return status OK	input arguments, APDU payload	APDU payload, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Remote	User Client (UC) - KSESSION_SSH: E - RAKME: G,E - RAKMA: G,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	module and the Smartcard				Admin session key establishment nCore lib: Remote Admin messaging	- KWARN pub: E
nCoreAPI Destroy	Remove handle to an object in RAM. If the current handle is the only one remaining, the object is zeroised from RAM	return status OK	input arguments, key handle	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KA: Z - LTX: Z
nCoreAPI Report stats	Reports the values of the statistics tree	return status OK	input arguments	statistics, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI New Enquiry (Show Status)	Report status information	return status OK	input arguments	status and versioning info, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E
nCoreAPI Set ACL	Replaces the ACL of a given key with a new ACL	return status OK	input arguments, key handle, acl	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KA: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
nCoreAPI Set app data	Writes the application information field of a key	return status OK	input arguments, key handle, app data	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH: E - KA: W
nCoreAPI Set NSO permissions	Sets the NSO key hash and which permissions require a Delegation Certificate	return status OK	input arguments, hash of KNSO (HKNSO), permissions	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	nShield Security Officer (NSO) - KSESSION_SSH: E - HKNSO: W
nCoreAPI Signature generation	Generate a digital signature or MAC value	return status OK	input arguments, mechanism, key handle, data to be signed	signed data, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Digital signature nCore lib: Message authentication	User Client (UC) - KSESSION_SSH: E - KA: E - KNSO: E
nCoreAPI Sign module state	Returns a signed certificate that contains data about the current configuration of the module	return status OK	input arguments	certificate, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Digital signature	User Client (UC) - KSESSION_SSH: E - KML: E
nCoreAPI Signature verification	Verifies a digital signature or MAC value	return status OK	input arguments, mechanism, key handle,	return status code	SSH lib: Symmetric Ciphers SSH lib: Message	User Client (UC) - KSESSION_SSH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			data to be verified		Authentication nCore lib: Digital signature nCore lib: Message authentication	: E - KA: E
nCoreAPI Write file	Writes a file to a Smartcard or Softcard	return status OK	input arguments, file info, slot, data	return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	nShield Security Officer (NSO) - KSESSION_SSH : E
nCoreAPI Write share	Writes a Share to a Smartcard or Softcard	return status OK	input arguments, file info, slot	data, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication nCore lib: Smartcard shares	User Client (UC) - KSESSION_SSH : E - SHAREKE: G,E - SHAREKM: G,E - LTX: E - KM: E
nCoreAPI SEE connection	Opens a connection from the nCoreAPI service to the SEE machine	return status OK	input arguments, uuid	world id, return status code	SSH lib: Symmetric Ciphers SSH lib: Message Authentication	User Client (UC) - KSESSION_SSH : E

Table 13: Approved Services

4.4 Non-Approved Services

N/A for this module.

4.5 External Software/Firmware Loaded

The nShield 5s cryptographic module's executable code is delivered by Entrust as a single signed firmware package (.npkg file).

5 Software/Firmware Security

5.1 Integrity Techniques

At start up, the following integrity tests are performed:

- The bootloader integrity is verified using RSA with 4096 bit key and SHA2-256, using NSBIK public key.
- The firmware integrity is verified using RSA with 4096 bit key and SHA2-256 using NFIK public key.
- The library partition integrity is verified using ECDSA with curve P-521 and SHA2-512 using NLIK public key.

5.2 Initiate on Demand

The operator can initiate the integrity tests on demand by rebooting the cryptographic module.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Hard and opaque epoxy potting	Monthly	The module should be inspected periodically for evidence of tamper attempts, including the entire enclosure including the epoxy resin security coating for obvious signs of damage

Table 14: Mechanisms and Actions Required

7.2 EFP/EFT Information

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature	2ºC	efp	shutdown
HighTemperature	95ºC	efp	shutdown
LowVoltage	8V	efp	shutdown
HighVoltage	14.5V	efp	shutdown

Table 15: EFP/EFT Information

7.3 Hardness Testing Temperature Ranges

Temperature Type	Temperature
LowTemperature	0ºC
HighTemperature	95ºC

Table 16: Hardness Testing Temperatures

8 Non-Invasive Security

Not applicable.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
Flash	Flash memory	Static
FRAM	Ferroelectic RAM	Static
RAM	Random Access Memory	Dynamic

Table 17: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Key blob load	External	RAM	Encrypted	Automated	Electronic	nCore lib: Key Blobs
Key blob make	RAM	External	Encrypted	Automated	Electronic	nCore lib: Key Blobs
Logical token load	External	RAM	Encrypted	Automated	Electronic	nCore lib: Smartcard shares
Logical token create	RAM	External	Encrypted	Automated	Electronic	nCore lib: Smartcard shares
SSH client key load	External	Flash	Encrypted	Automated	Electronic	
FW keys load	External	Flash	Plaintext	Automated	Electronic	nCore lib: FW integrity

Table 18: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Factory Reset	setup factorystate command zeroises all unprotected SSPs	All unprotected SSPs are zeroised, overwritten with zeroes	Operator Initiated

Table 19: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KRESET	Resettable key	256 bits - 256 bits	KDF secret - CSP	Platform RBG		nCore lib: KDFs
KUSER_SSH	Global SSH key encryption key	256 bits - 256 bits	AES key - CSP	nCore lib: KDFs		nCore lib: Symmetric Ciphers
KSSH_SETUP	Server authentication SSH private key for the Setup service	P-256 - 128 bits	ECDSA private key - CSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KSSH_UPDATER	Server authentication SSH private key for the Updater service	P-256 - 128 bits	ECDSA private key - CSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KSSH_SSHADMIN	Server authentication SSH private key for the SSHAdmin service	P-256 - 128 bits	ECDSA private key - CSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KSSH_MONITOR	Server authentication SSH private key for the Monitor service	P-256 - 128 bits	ECDSA private key - CSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KSSH_LAUNCHER	Server authentication SSH private key for the Launcher service	P-256 - 128 bits	ECDSA private key - CSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KSSH_SETUP pub	Server authentication SSH public key for the Setup service	P-256 - 128 bits	ECDSA public key - PSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KSSH_UPDATER pub	Server authentication SSH public key for the Updater service	P-256 - 128 bits	ECDSA public key - PSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KSSH_SSHADMIN pub	Server authentication SSH public key for the SSHAdmin service	P-256 - 128 bits	ECDSA public key - PSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KSSH_MONITOR pub	Server authentication SSH public key for the Monitor service	P-256 - 128 bits	ECDSA public key - PSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KSSH_LAUNCHER pub	Server authentication SSH public key for the Launcher service	P-256 - 128 bits	ECDSA public key - PSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KSESSION_SSH	SSH channel session keys	128 bits - 128 bits	Symmetric key - CSP		SSH lib: Key Agreement	SSH lib: Symmetric Ciphers SSH lib: Message Authentication
NPSK pub	Firmware package public signature	P-521 - 256 bits	ECDSA public key - PSP	ECDSA KeyGen (FIPS186-4) (A3707)		nCore lib: FW integrity

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	verification key					
DSK pub	Developer public signing key	P-521 - 256 bits	ECDSA public key - PSP			nCore lib: Digital signature
ESK pub	Root Entrust SEE public signing key	P-521 - 256 bits	ECDSA public key - PSP			nCore lib: Digital signature
KSSH_CLIENT pub	Client authentication SSH public key for the Updater, Setup, SSHAdmin, nCoreAPI, Monitor, Launcher service	P-256 - 128 bits	ECDSA public key - PSP			SSH lib: Digital Signature
DRBG-1 entropy input	Entropy input for the DRBG in the Platform RBG	520 bits - >= 256 bits	DRBG Entropy Input - CSP	Platform RBG		Platform RBG
DRBG-1 seed, Internal State: C and V values	Internal state for the DRBG in the Platform RBG	256 bits - 256 bits	DRBG Internal State - CSP	Platform RBG		Platform RBG
KCONTAINER	Master key for container	256 bits - 256 bits	KDF secret - CSP	nCore lib: KDFs		nCore lib: KDFs
KCONTAINERSSH	Encryption key for KSSH_NCORE	256 bits - 256 bits	AES key - CSP	nCore lib: KDFs		nCore lib: Symmetric Ciphers
KSSH_NCORE	Server authentication SSH private key for the nCoreAPI service	P-256 - 128 bits	ECDSA private key - CSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KSSH_NCORE pub	Server authentication SSH public key for the nCoreAPI service	P-256 - 128 bits	ECDSA public key - PSP	ECDSA KeyGen (FIPS186-4) (A3707)		SSH lib: Digital Signature
KRE_BLOBKEY	Security World recovery confidentiality key	3072 bits - 128 bits	RSA private key - CSP	RSA KeyGen (FIPS186-4) (A3707)		nCore lib: Digital signature
KR	Security World recovery key	256 bits - 256 bits	AES key - CSP	nCoreAPI RBG		nCore lib: KDFs
IMPATHKE	Confidentiality session key for Impath channel	256 bits - 256 bits	AES key - CSP		nCore lib: Impath session key establishment	nCore lib: Impath messaging
IMPATHKM	Integrity session key for Impath channel	256 bits - 256 bits	HMAC key - CSP		nCore lib: Impath session key establishment	nCore lib: Impath messaging
KA	Security World application keys	>=112 bits - >=112 bits	any application key - CSP	nCoreAPI RBG DSA KeyGen (FIPS186-4) (A3707) ECDSA KeyGen (FIPS186-4) (A3707) KDF SP800-108 (A3707) RSA KeyGen	KAS-ECC Sp800-56Ar3 (A3707) KTS-IFC (A3707)	nCore lib: Symmetric Ciphers nCore lib: Digital signature nCore lib: Message authentication nCore lib: Secure hashing

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
				(FIPS186-4) (A3707)		
KM	Security World module key	256 bits - 256 bits	AES key - CSP	nCoreAPI RBG		KDF SP800-108 (A3707)
KML	Security World module signing key	3072 bits - 128 bits	DSA key - CSP	nCoreAPI RBG		nCore lib: Digital signature
KNSO	Security World NSO key	3072 bits - 128 bits	DSA key - CSP	nCoreAPI RBG		nCore lib: Digital signature
HKNSO	Security World hash of public KNSO	256 bits - 256 bits	SHA-256 hash - PSP	nCore lib: Secure hashing		
BLOBKE	Security World key blob encryption key	256 bits - 256 bits	AES key - CSP		KDF SP800-108 (A3707)	nCore lib: Symmetric Ciphers
BLOBKM	Security World key blob integrity key	256 bits - 256 bits	HMAC key - CSP		KDF SP800-108 (A3707)	nCore lib: Message authentication
LTX	Security World Logical Token for key X	256 bits - 256 bits	KDF secret - CSP			KDF SP800-108 (A3707)
SHAREKE	Security World Share encryption key	256 bits - 256 bits	AES key - CSP		nCore lib: Smartcard shares	nCore lib: Smartcard shares
SHAREKM	Security World Share integrity key	256 bits - 256 bits	HMAC key - CSP		nCore lib: Smartcard shares	nCore lib: Smartcard shares

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
RAKME	Remote Admin channel encryption session key	256 bits - 256 bits	AES key - CSP		nCore lib: Remote Admin session key establishment	nCore lib: Remote Admin messaging
RAKMA	Remote Admin channel encryption integrity key	256 bits - 256 bits	CMAC key - CSP		nCore lib: Remote Admin session key establishment	nCore lib: Remote Admin messaging
KAL	Security World audit logging key	3072 bits - 128 bits	DSA key - CSP	DSA KeyGen (FIPS186-4) (A3707)		DSA SigGen (FIPS186-4) (A3707)
KWARN pub	Entrust root warranting public key for Administrator cards and Operator cards	P-521 - 256 bits	ECDSA key - PSP			ECDSA SigVer (FIPS186-4) (A3707)
DRBG-2 entropy input	Entropy input for the DRBG in the nCoreAPI service RBG	520 bits - >= 256 bits	DRBG Entropy Input - CSP	Platform RBG		nCoreAPI RBG
DRBG-2 seed, Internal State: C and V values	Internal state for the DRBG in the Platform RBG	256 bits - 256 bits	DRBG Internal State - CSP	nCoreAPI RBG		nCoreAPI RBG

Table 20: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
KRESET		FRAM:Plaintext		Factory Reset	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
KUSER_SSH		RAM:Plaintext	Until reboot	Factory Reset	KRESET:derived from
KSSH_SETUP		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSSH_UPDATER		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSSH_SSHADMIN		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSSH_MONITOR		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSSH_LAUNCHER		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSSH_SETUP pub		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSSH_UPDATER pub		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSSH_SSHADMIN pub		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSSH_MONITOR pub		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSSH_LAUNCHER pub		Flash:Encrypted		Factory Reset	KUSER_SSH:encrypted by
KSESSION_SSH		RAM:Plaintext	Until reboot or channel closure		
NPSK pub	FW keys load	Flash:Plaintext			
DSK pub	SSH client key load	Flash:Plaintext		Factory Reset	
ESK pub	FW keys load	Flash:Plaintext		Factory Reset	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
KSSH_CLIENT pub	SSH client key load	Flash:Plaintext		Factory Reset	
DRBG-1 entropy input		RAM:Plaintext	Until seed is derived		
DRBG-1 seed, Internal State: C and V values		RAM:Plaintext	Until reboot or reseed		
KCONTAINER		RAM:Plaintext	Until reboot	Factory Reset	KRESET:derived from
KCONTAINERSSH		RAM:Plaintext	Until reboot	Factory Reset	KCONTAINER:derived from
KSSH_NCORE		Flash:Encrypted		Factory Reset	KCONTAINERSSH:encrypted by
KSSH_NCORE pub		Flash:Encrypted		Factory Reset	KCONTAINERSSH:encrypted by
KRE_BLOBKEY	Key blob load Key blob make	RAM:Plaintext		Factory Reset	LTX:encrypted by
KR	Key blob load Key blob make	RAM:Plaintext		Factory Reset	KRE_BLOBKEY:encrypted by
IMPATHKE		RAM:Plaintext	Until reboot or channel closure		
IMPATHKM		RAM:Plaintext	Until reboot or channel closure		

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
KA	Key blob load Key blob make	RAM:Plaintext	Until key handle destroyed	Factory Reset	LTX:encrypted by
KM	Key blob load Key blob make	Flash:Plaintext		Factory Reset	LTX:encrypted by
KML		Flash:Plaintext		Factory Reset	
KNSO	Key blob load Key blob make	Flash:Plaintext	Until key handle destroyed	Factory Reset	LTX:encrypted by
HKNSO		Flash:Plaintext		Factory Reset	
BLOBKE		RAM:Plaintext	Until key handle destroyed	Factory Reset	LTX:derived from
BLOBKM		RAM:Plaintext		Factory Reset	LTX:derived from
LTX	Logical token load Logical token create	RAM:Plaintext	Until key handle destroyed	Factory Reset	
SHAREKE		RAM:Plaintext	Until key handle destroyed	Factory Reset	
SHAREKM		RAM:Plaintext	Until key handle	Factory Reset	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			destroyed		
RAKME		RAM:Plaintext	Until session closure or reboot	Factory Reset	
RAKMA		RAM:Plaintext	Until session closure or reboot	Factory Reset	
KAL		Flash:Plaintext		Factory Reset	
KWARN pub	FW keys load	Flash:Plaintext		Factory Reset	
DRBG-2 entropy input		RAM:Plaintext	Until seed is derived		
DRBG-2 seed, Internal State: C and V values		RAM:Plaintext	Until reboot or reseed		

Table 21: SSP Table 2

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
RSA SigVer (FIPS 186-4)	4096 bit	Signature verification	SW/FW Integrity	status output	Verify
ECDSA SigVer (FIPS 186-4)	P-521	Signature verification	SW/FW Integrity	status output	Verify

Table 22: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SP 800-90B health tests (APT, RCT)	-	-	CAST	Successful initialization of the module	SP 800-90B Section 4	Continuous
nCore lib: DSA keyGen	2048 bits	PCT	PCT	n/a	Signature generation and verification	Key generation
nCore lib: ECDSA keyGen	P-224	PCT	PCT	n/a	Signature generation and verification	Key generation
nCore lib: RSA keyGen	2048 bits	PCT	PCT	n/a	Key generation	Key generation
nCore lib: Safe prime keyGen	MODP-2048	PCT	PCT	n/a	Safe prime generation and verification	Safe prime generation
Firmware load test	ECDSA sigVer, P-521	Load Test	SW/FW Load	n/a	Firmware load test	Upgrading firmware
nCore lib: AES-ECB	128, 192, 256 bits	KAT	CAST	Successful initialization of the module	encrypt/decrypt	Module Initialization
nCore lib: AES-CMAC	128 bits	KAT	CAST	Successful initialization	MAC generation and verification	Module Initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				of the module		
nCore lib: HMAC	HMAC with SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	KAT	CAST	Successful initialization of the module	MAC generation and verification	Module Initialization
nCore lib: SHA-1		KAT	CAST	Successful initialization of the module		Module Initialization
nCore lib: SHA-3	SHA3-224, SHA3-256, SHA3-384, SHA3-512	KAT	CAST	Successful initialization of the module		Module Initialization
nCore lib: RSASSA-PKCS-v1_5	2048 bits, SHA2	KAT	CAST	Successful initialization of the module	Signature generation and verification	Module Initialization
nCore lib: RSA-OAEP	2048 bits	KAT	CAST	Successful initialization of the module	encrypt/decrypt	Module Initialization
nCore lib: DSA	2048 bits, SHA2-224	KAT	CAST	Successful initialization of the module	Signature generation and verification	Module Initialization
nCore lib: ECDSA	P-224, B-233	KAT	CAST	Successful initialization of the module	Signature generation and verification	Module Initialization
nCore lib: Pair-Wise Consistency test		PCT	CAST	n/a		Key generation
nCore lib: KAS-FFC	MODP-2048, MODP-3072	KAT	CAST	Successful initialization of the module	Shared secret computation	Module Initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
nCore lib: KAS-ECC	P-224, B-233	KAT	CAST	Successful initialization of the module	Shared secret computation	Module Initialization
nCore lib: One-step KDF	SHA2-256 aux function	KAT	CAST	Successful initialization of the module	Key derivation	Module Initialization
nCore lib: Two-step KDF	HMAC-SHA2-256 aux function	KAT	CAST	Successful initialization of the module	Key derivation	Module Initialization
nCore lib: KBKDF	Counter KDF with CMAC-AES256	KAT	CAST	Successful initialization of the module	Key derivation	Module Initialization
nCore lib: DRBG	Instantiate, reseed, generate as per SP 800-90Arev1 section 11.3	KAT	CAST	Successful initialization of the module	Key derivation	Module Initialization
SSH lib: AES-GCM	128 bits	KAT	CAST	Successful initialization of the module	encrypt/decrypt	Module Initialization
SSH lib: AES-CTR	128 bits	KAT	CAST	Successful initialization of the module	encrypt/decrypt	Module Initialization
SSH lib: HMAC	HMAC-SHA2-256 and HMAC-SHA2-512	KAT	CAST	Successful initialization of the module	MAC generation and verification	Module Initialization
SSH lib: KAS ECC	ECDH P-256 shared secret computation	KAT	CAST	Successful initialization of the module	MAC generation and verification	Module Initialization
SSH lib: ECDSA	Sign/Verify P-256 and P-521	KAT	CAST	Successful initialization	Signature generation and verification	Module Initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				of the module		
SSH lib: SSH KDF		KAT	CAST	Successful initialization of the module	MAC generation and verification	Module Initialization
SSH lib: ECDH PCT		PCT	CAST	n/a		Key generation
Bootloader: RSASSA-PKCS-v1_5	4096 bits, SHA2-256	KAT	CAST	Successful initialization of the module	Signature verification	Module Initialization

Table 23: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS 186-4)	Signature verification	SW/FW Integrity	User initiated module reboot	n/a
ECDSA SigVer (FIPS 186-4)	Signature verification	SW/FW Integrity	User initiated module reboot	n/a

Table 24: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SP 800-90B health tests (APT, RCT)	-	CAST	Continuous Every 24h	Automatic
nCore lib: DSA keyGen	PCT	PCT	n/a	n/a
nCore lib: ECDSA keyGen	PCT	PCT	n/a	n/a
nCore lib: RSA keyGen	PCT	PCT	n/a	n/a
nCore lib: Safe prime keyGen	PCT	PCT	n/a	n/a
Firmware load test	Load Test	SW/FW Load	n/a	n/a

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
nCore lib: AES-ECB	KAT	CAST	Every 24h	Automatic
nCore lib: AES-CMAC	KAT	CAST	Every 24h	Automatic
nCore lib: HMAC	KAT	CAST	Every 24h	Automatic
nCore lib: SHA-1	KAT	CAST	Every 24h	Automatic
nCore lib: SHA-3	KAT	CAST	Every 24h	Automatic
nCore lib: RSASSA-PKCS-v1_5	KAT	CAST	Every 24h	Automatic
nCore lib: RSA-OAEP	KAT	CAST	Every 24h	Automatic
nCore lib: DSA	KAT	CAST	Every 24h	Automatic
nCore lib: ECDSA	KAT	CAST	Every 24h	Automatic
nCore lib: Pair-Wise Consistency test	PCT	CAST		Automatic
nCore lib: KAS-FFC	KAT	CAST	Every 24h	Automatic
nCore lib: KAS-ECC	KAT	CAST	Every 24h	Automatic
nCore lib: One-step KDF	KAT	CAST	Every 24h	Automatic
nCore lib: Two-step KDF	KAT	CAST	Every 24h	Automatic
nCore lib: KBKDF	KAT	CAST	Every 24h	Automatic
nCore lib: DRBG	KAT	CAST	Every 24h	Automatic
SSH lib: AES-GCM	KAT	CAST	Every 24h	Automatic
SSH lib: AES-CTR	KAT	CAST	Every 24h	Automatic
SSH lib: HMAC	KAT	CAST	Every 24h	Automatic
SSH lib: KAS ECC	KAT	CAST	Every 24h	Automatic
SSH lib: ECDSA	KAT	CAST	Every 24h	Automatic
SSH lib: SSH KDF	KAT	CAST	Every 24h	Automatic
SSH lib: ECDH PCT	PCT	CAST		Automatic

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Bootloader: RSASSA-PKCS-v1_5	KAT	CAST	n/a	n/a

Table 25: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	The module's error state.	POST or CAST failure	Reboot Module	Status LED

Table 26: Error States

10.5 Operator Initiation of Self-Tests

Operators can initiate the integrity tests on demand by restarting the module.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The nShield cryptographic module is sent to the customers using a standard carrier service. After accepting the delivery of the module, a physical inspection of the module shall be performed (refer to Physical Security section). This inspection is done to ensure that the module has not been tampered with during transit. If the inspection results indicate that the module has not been tampered with, the Administrator can then proceed with installation and configuration of the module.

The cryptographic module supports firmware upgrades in the field, which are provided by Entrust as a single signed firmware package (.npkg file).

This following instructions describe how to inspect the cryptographic module´s firmware and hardware version information and ensure they correspond with the FIPS 140-3 validated versions.

Firmware identification

The cryptographic module provides the service updater info which provides firmware version information in JSON format. Entrust provides the `hsmadmin status` command-line utility which calls the service updater info internally.

```
hsmadmin status –json

{
    "D5DE-E1F8-D6E7": {
        "succeeded": true,
        "data": {
            "mode": "primary",
            "primary-version": "13.4.5-751-56c6f1db",
            "recovery-version": "13.2.4-280-7f4f0c24",
            "uboot-version": "1.1.0-1245-b9bedfa"
        }
    }
}
```

The following fields in the output must be checked:

Field	Expected value
primary-version	13.4.5-751-56c6f1db
recovery-version	13.2.4-280-7f4f0c24
uboot-version	1.1.0-1245-b9bedfa or 1.4.1-0-edb84d6e or 14.2.0-0-23012d8

Hardware identification

The cryptographic module provides the command `Cmd_NewEnquiry` which reports hardware version information.

Entrust provides the `enquiry` command-line utility which calls `Cmd_NewEnquiry` internally

```
product name nC5536E
hardware part no PCA10005-01 revision 03
```

The following fields in the output must be checked:

Field	Expected value
product name	nC5536E or nC5536N
hardware part no	PCA10005-01 revision 03 or revision 04

Alternatively, the cryptographic module also provides the service `setup info` which provides hardware version information in JSON format. Entrust provides the `hsmadmin info` command-line utility which calls the service setup info internally.

```
hsmadmin info –json

{
"15C8-4387-C748": {
"eeprom": {
...
,
"buildpart":
{ "value": "PCA10005-01", "crc": xxxxx }
,
"buildrev":
{ "value": "03", "crc": xxxxx }
...
}
}
```

The following fields in the output must be checked:

Field	Expected value
buildpart	"value": "PCA10005-01"
buildrev	"value": "03" or "value": "04"

11.2 Administrator Guidance

The NSO shall initialize the Security World in Approved mode.

11.3 Non-Administrator Guidance

Users shall verify that the cryptographic module is initialized and configured in Approved mode of operation.

12 Mitigation of Other Attacks

Not applicable.