ENTRUST

SECURING A WORLD IN MOTION

nShield 5s Hardware Security Module

FIPS 140-3 Level 3 non-proprietary Security Policy

Version: 1.0.5

Date: 08/01/2025

Information in this document is subject to change without notice.

nCipher Security Limited makes no warranty of any kind with regard to this information, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. nCipher Security Limited shall not be liable for errors contained herein or for incidental or consequential damages concerned with the furnishing, performance or use of this material.

Where translations have been made in this document English is the canonical language.

nCipher Security Limited
Registered Office: One Station Square,
Cambridge, CB1 2GA, United Kingdom
Registered in England No. 11673268

nCipher is an Entrust company.

Because we are continuously improving our products and services, Entrust Corporation reserves the right to change specifications without prior notice. Entrust is an equal opportunity employer.

Contents

1 General

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**. The following table specifies the security level in detail.

ISO/IEC 24759 Section 6. **[Number Below]**	**FIPS 140-3 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

Table 1 Security levels

2 Cryptographic module specification

2.1 Scope

The following product hardware variants and firmware version(s) are in scope of this Security Policy.

Model	Hardware [Part Number and Version]	Firmware Version	Distinguishing Features
nShield 5s F3 model number nC5536E	PCB Assembly Part Number: PCA10005-01 PCB Assembly Revision: 03, 04	primary: 13.2.4 recovery: 13.2.4 uboot: 1.1.0, 1.4.1	PCIe form factor
nShield 5s for nShield 5c and for nShield HSMi model number nC5536N			PCIe form factor identical to the nShield 5s F3 (nC5536E), embedded inside the nShield 5c or the nShield HSMi network appliances.

Table 2 Cryptographic Module Tested Configuration

2.2 Cryptographic module description

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

Figure 1 nShield 5s (left) and nShield 5c (right)

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.

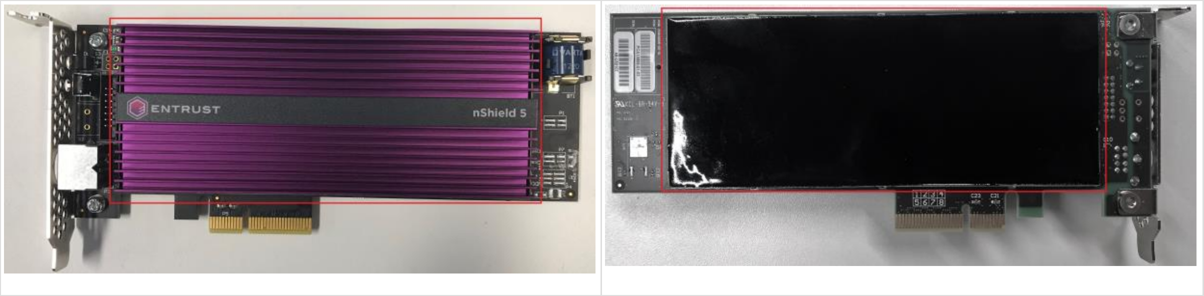

Figure 2 Cryptographic module physical boundary

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

2.3 Supported cryptographic algorithms

This section describes the cryptographic mechanisms and security functions provided and used by the cryptographic module.

2.3.1 Approved algorithms

The following tables describe the approved cryptographic algorithms supported by the Cryptographic Module.

Note: All AES symmetric key sizes have equivalent strength.

2.3.1.1 nCore crypto

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A2513	AES [FIPS 197] [SP 800-38A] [SP 800-38D]	ECB CBC GCM	Key sizes: 128, 192, 256 bits.	Data encryption/decryption
A2513	KTS (AES) [SP 800-38F] [SP 800-38D]	KW KWP GCM	Key sizes: 128, 192, 256 bits. (Key establishment methodology provides between 128, 192, 256 bits of encryption strength.)	Key wrapping/unwrapping
A2513	KTS-IFC [SP 800-56Brev2]	KTS-IFC (KTS-OAEP-basic) with SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	Modulus length: 2048, 3072, 4096 bits. (Key establishment methodology provides between 112 and 150 bits of encryption strength.)	Key transport (encapsulation, un-encapsulation)
Vendor affirmed	CKG [SP 800-133rev2]	Section 6.1 "Direct Generation" of symmetric keys	n/a	Symmetric key generation

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A2513	RSA keyGen [FIPS 186-4]	RSASSA-PKCS-v1_5, RSASSA-PSS	Modulus length: 2048, 3072, 4096 bits. (Provides 112, 128, 150 bits of strength.)	Key generation
A2513	RSA SigGen [FIPS 186-4]	RSASSA-PKCS-v1_5, RSASSA-PSS with SHA2-224, SHA2-256, SHA2-384, SHA2-512	Modulus length: 2048, 3072, 4096 bits. (Provides 112, 128, 150 bits of strength.)	Signature generation
A2513	RSA SigVer [FIPS 186-4]	RSASSA-PKCS-v1_5, RSASSA-PSS with SHA-1 (legacy use only), SHA2-224, SHA2-256, SHA2-384, SHA2-512	Modulus length: 1024 (legacy use only), 2048, 3072, 4096 bits.	Signature verification
A2513	ECDSA KeyGen [FIPS 186-4]	n/a	Curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571 (Provides between 112 and 256 bits of encryption strength.)	Key generation
A2513	ECDSA KeyVer [FIPS 186-4]	n/a	Curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571 (Provides between 112 and 256 bits of encryption strength.)	Key verification
A2513	ECDSA SigGen [FIPS 186-4]	n/a	Curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571 (Provides between 112 and 256 bits of encryption strength.)	Signature generation
A2513	ECDSA SigVer [FIPS 186-4]	n/a	Curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571	Signature verification

A2513	DSA KeyGen
	[FIPS 186-4]
		n/a	L = 2048 bits, N = 224 bits
			L = 2048 bits, N = 256 bits
			L = 3072 bits, N = 256 bits
			(Provides between 112 and 128 bits of strength.)
				Key generation

A2513	DSA SigGen
	[FIPS 186-4]
		Uses SHA2-224, SHA2-256, SHA2-384, SHA2-512	L = 2048 bits, N = 224 bits
			L = 2048 bits, N = 256 bits
			L = 3072 bits, N = 256 bits
			(Provides between 112 and 128 bits of strength.)
				Signature generation

A2513	DSA SigVer
	[FIPS 186-4]
		Uses SHA-1 (legacy use only), SHA2-224, SHA2-256, SHA2-384, SHA2-512	L = 1024 bits, N = 160 bits (legacy use only)
			L = 2048 bits, N = 224 bits
			L = 2048 bits, N = 256 bits
			L = 3072 bits, N = 256 bits
				Signature verification

A2513	DSA PQGGen	n/a	L = 2048 bits, N = 224 bits	Domain parameter generation

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function

	[FIPS 186-4]		L = 2048 bits, N = 256 bits
			L = 3072 bits, N = 256 bits
			(Provides between 112 and 128 bits of strength.)	

A2513	DSA PQGVer
	[FIPS 186-4]
		n/a	L = 1024 bits, N = 160 bits (legacy use only)
			L = 2048 bits, N = 224 bits
			L = 2048 bits, N = 256 bits
			L = 3072 bits, N = 256 bits
				Domain parameter verification

A2513	HMAC [FIPS 198]	HMAC-SHA1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512	Key sizes: 128 bits =< key size =< 2048 bits. Output length: 10 to 64 bytes. (Provides between 128 and 256 bits of strength.)	MAC generation and verification
A2513	AES [SP 800-38B]	CMAC	Key sizes: 128, 192, 256 bits.	MAC generation and verification
A2513	KAS-FFC [SP 800-56Arev3]	DH	Domain params: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192, FB, FC KDF: oneStep with auxiliary functions SHA2-224, SHA2-256, SHA2-384, SHA2-512 (Key establishment methodology provides between 112 and 200 bits of encryption strength.)	Key agreement, compliant with IG D.F. Scenario 2, path 2.
A2513	KAS-FFC [SP 800-56Arev3]	DH	Domain params: MODP-3072 KDF: oneStep with auxiliary function SHA2-256 (Key establishment methodology provides 128 bits of encryption strength.)	Key agreement used in Impath channel, compliant with IG D.F. Scenario 2, path 2.
A2513	Safe Primes Key Generation [SP 800-56Arev3]	KeyGen for KAS-FFC	Safe prime groups: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 (Provides between 112 and 200 bits of encryption strength.)	KAS-FFC key generation
A2513	Safe Primes Key Verification [SP 800-56Arev3]	KeyVer for KAS-FFC	Safe prime groups: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	KAS-FFC key verification
A2513	KAS-ECC [SP 800-56Arev3]	ECDH ECMQV	Curves: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571	Key agreement, compliant with IG D.F. Scenario 2, path 2.

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
			KDF: oneStep with auxiliary functions SHA2-224, SHA2-256, SHA2-384, SHA2-512 (Key establishment methodology provides between 112 and 256 bits of encryption strength.)	
A2513	KAS-ECC [SP 800-56Arev3]	ECDH	Curves: P-521 KDF: twoStep with auxiliary HMAC-SHA2-256 (Key establishment methodology provides 256 bits of encryption strength.)	Key agreement for smartcard channel, compliant with IG D.F. Scenario 2, path 2.
A2513	KBKDF [SP 800-108rev1]	counter mode CMAC-AES256	Key sizes: 128, 192, 256 bits. Output length: 128 bits. (Provides between 128 and 256 bits of strength.)	Key derivation
A2513	SHS [FIPS 180-4]	SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	n/a	Message digest
A2513	SHA-3 [FIPS 202]	SHA3-224, SHA3-256, SHA3-384, SHA3-512	n/a	Message digest
A2513	DRBG [SP 800-90Arev1]	Hash_DRBG	256 bits of security strength	Random bit generation

Table 3 nCore - Approved Algorithms

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

Note: 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.

2.3.1.2 Bootloader crypto

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A2404 A6385	RSA SigVer [FIPS 186-4]	RSASSA-PKCS-v1_5 with SHA2-256	Modulus length: 4096 bits (Provides 149 bits of strength)	Signature verification
A2404 A6385	SHS [FIPS 180-4]	SHA2-256	n/a	Message digest

Table 4 Bootloader - Approved Algorithms

2.3.1.3 SSH crypto

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A2512	AES [FIPS 197] [SP 800-38A] [SP 800-38D]	CTR GCM ECB	Key sizes: 128 bits	Data encryption/decryption
Vendor affirmed	CKG [SP 800-133rev2]	Section 6.1 "Direct Generation" of symmetric keys	n/a	Symmetric key generation
A2512	ECDSA SigGen [FIPS 186-4]	n/a	Curves: P-256, P-521 (Provides between 128 and 256 bits of strength.)	Signature generation
A2512	ECDSA SigVer [FIPS 186-4]	n/a	Curves: P-256, P-521 (Provides between 128 and 256 bits of strength.)	Signature verification
A2512	HMAC [FIPS 198]	HMAC-SHA2-256	Key sizes: 256 bits Output length: 32 bytes (Provides 256 bits of strength.)	MAC generation and verification
A2512	KAS-ECC-SSC [SP 800-56Arev3]	ECDH	Curves: P-256 (Key establishment methodology provides 128 bits of encryption strength)	Key agreement
A2512	CVL - Secure Shell (SSHv2) KDF [SP 800-135rev1]	n/a	n/a	Key derivation
A2512	KAS-ECC [SP 800-56Arev3] [SP 800-135rev1]	ECDH	Curves: P-256 KDF (CVL): SSHv2 (Key establishment methodology provides 128 bits of encryption strength)	Key agreement, KAS-ECC-SSC (Cert. #A2512) in conjunction with SSHv2 KDF (CVL) of SP 800-135rev1 (Cert. #A2512), which is compliant with IG D.F. Scenario 2, path 2.
A2512	SHS [FIPS 180-4]	SHA2-256 SHA2-512	n/a	Message digest

Table 5 SSH - Approved Algorithms

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

Note: For AES GCM, 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.

2.3.2 Allowed algorithms

The following table describes the allowed cryptographic algorithms supported by the Cryptographic Module.

2.3.2.1 nCore crypto

Algorithm	Caveat	Use/Function
ECDSA [FIPS 186-4]	• 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)	Key generation Signature generation and verification
KAS-ECC [SP 800-56Arev3]	• 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)	Key agreement

Table 6 nCore - Non-Approved Algorithms Allowed in the Approved Mode of Operation

2.3.3 Non-approved algorithms

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

3 Cryptographic module interfaces

The Cryptographic Module provides the following physical ports:

- Status LED
- Recovery button
- Smartcard reader serial port
- Host interface PCIe bus
- Battery (including external backup battery power supply)

The following table maps the FIPS logical interfaces and physical ports to the module´s services

Physical port	**Logical interface**	**Data that passes over port/interface**
PCIe bus	Data input	nCoreAPI, Updater, SSHAdmin
	Data output	nCoreAPI, SSHAdmin
	Control input	nCoreAPI, SSHAdmin, Setup, Monitor, Discovery
	Status output	nCoreAPI, Updater, Setup, Monitor, Discovery
	Power	n/a
Smartcard reader serial port	Data input	APDU commands
	Data output	APDU commands
Status LED	Status output	n/a
Recovery button	Control input	n/a
Battery	Power	n/a

Table 7 Ports and Interfaces

Note: Control output is omitted because the module does not implement it.

The following figure shows the module’s physical ports:

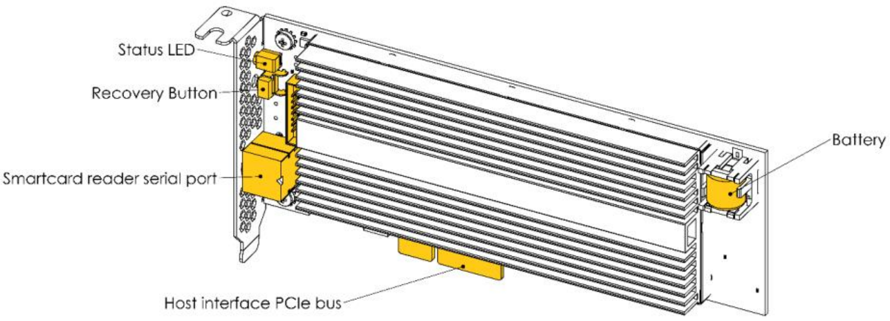

Figure 3 Physical ports

4 Roles, services and authentication

4.1 Roles

The Cryptographic Module supports the following roles:

- Platform Crypto Officer (PCO)
- nShield Security Officer (NSO)
- User Client (UC)

Platform Crypto Officer (PCO)

This role is responsible of administration tasks of the HSM platform.

To assume this role, an operator needs to open a session with the SSHAdmin, Updater, Setup or Monitor services, using its SSH private key for each service.

When the module is in factory state, the SSHAdmin client SSH keys are set to a default value. These keys must be changed with the `SSHAdmin set` service before the module can be initialized and used.

nShield Security Officer (NSO)

This role is represented by Administrator Card holders, which have access to KNSO and are responsible for the overall management of a Security World.

To assume this role, an operator or group of operators need to present a quorum *m of N* of smartcards, and the KNSO Key Blob. Each operator is identified by its individual smartcard, which contains a unique logical token share. NSO smartcard sets are initialized at Security World creation time.

User Client (UC)

This role is authorised to use the general purpose cryptographic services offered by the cryptographic module.

To assume this role, a client application needs to open a session with the nCoreAPI service, using its SSH private key. The client application's SSH public key must have been previously loaded into the module by the PCO using the SSHAdmin service.

Role	Service	Input	Output
PCO	setup info	Input arguments	Versioning information, return status code
PCO	setup factorystate	Input arguments	return status code
PCO	setup gettime	Input arguments	Time, return status code
PCO	setup settime	Input arguments, time	return status code
PCO	sshadmin set	Input arguments, service, SSH client public key	return status code
PCO	sshadmin list	Input arguments, service	SSH client public key, return status code

Role	Service	Input	Output
PCO	sshadmin get-serverkey	Input arguments, service	SSH server public key (KSSH_UPDATER, KSSH_SSHADMIN, KSSH_MONITOR, KSSH_SETUP), return status code
PCO	updater info	Input arguments	Versioning information, return status code
PCO	updater receive	Input arguments, fw update file	fw update file info, return status code
PCO	updater load	Input arguments, fw update file info	return status code
PCO	monitor getlog	Input arguments	Logs, return status code
PCO	monitor clearlog	Input arguments	return status code
All	discovery python-zeroconf	Input arguments	IP address, return status code
UC	nCoreAPI Big number operation	Input arguments	Operation result, return status code
UC	nCoreAPI Make Blob	Input arguments, key handle	Key blob, return status code
UC	nCoreAPI Bulk channel	Input arguments, data	Operation result, return status code
UC	nCoreAPI Check User Action	Input arguments, key handle	return status code
UC	nCoreAPI Clear Unit	Input arguments, module mode	return status code
NSO	nCoreAPI Set Module Key	Input arguments, key handle	return status code
NSO	nCoreAPI Remove Module Key	Input arguments, key hash	return status code
UC	nCoreAPI Duplicate key handle	Input arguments, key handle	Key handle, return status code
UC	nCoreAPI Enable feature	Input arguments, features	return status code
UC	nCoreAPI Encryption	Input arguments, mechanism, key handle, plaintext data, iv	Encrypted data, return status code
UC	nCoreAPI Decryption	Input arguments, mechanism, key handle, encrypted data	Decrypted data, return status code
UC	nCoreAPI Erase from smartcard /softcard	Input arguments, slot, file info	return status code
UC	nCoreAPI Format Token	Input arguments, slot	return status code
UC	nCoreAPI File operations	Input arguments, file info, operation	return status code
UC	nCoreAPI Force module to fail	Input arguments	return status code
UC	nCoreAPI Generate prime number	Input arguments, length	Bignumber, return status code
UC	nCoreAPI Generate random number	Input arguments, length	Random bytes, return status code
UC	nCoreAPI Get ACL	Input arguments, key handle	acl, return status code
UC	nCoreAPI Get key application data	Input arguments, key handle	Application data, return status code
UC	nCoreAPI Get challenge	Input arguments	Nonce, return status code

Role	Service	Input	Output
UC	nCoreAPI Get KLF2	Input arguments	Key handle, return status code
UC	nCoreAPI Get Key Information	Input arguments, key handle	Key info, return status code
UC	nCoreAPI Get module signing key	Input arguments	Key handle, return status code
UC	nCoreAPI Get list of slot in the module	Input arguments	Slots info, return status code
UC	nCoreAPI Get Logical Token Info	Input arguments, key handle	Logical token info, return status code
UC	nCoreAPI Get list of module keys	Input arguments	hash of KNSO (HKNSO), hash of module keys, return status code
UC	nCoreAPI Get module state	Input arguments	Module attributes, return status code
UC	nCoreAPI Get real time clock	Input arguments	time, return status code
UC	nCoreAPI Get share access control list	Input arguments, slot	acl, return status code
UC	nCoreAPI Get Slot Information	Input arguments, slot	smartcard info, return status code
UC	nCoreAPI Get Ticket	Input arguments	Ticket, return status code
UC	nCoreAPI Initialize Unit	Input arguments	return status code
UC	nCoreAPI Insert a Softcard	Input arguments, slot	return status code
UC	nCoreAPI Remove a Softcard	Input arguments, slot	return status code
UC	nCoreAPI Impath channel	Input arguments, data	data, return status code
UC	nCoreAPI Key generation	Input arguments, key params, acl, app data	Key handle, key generation cert, return status code
UC	nCoreAPI Key import	Input arguments, wrapped key, acl, app data	Key handle, return status code
UC	nCoreAPI Derive Key	Input arguments, mechanism, key handles	Key handle, return status code
UC	nCoreAPI Load Blob	Input arguments, blob data	Key handle, return status code
UC	nCoreAPI Load Logical Token	Input arguments, logical token hash	Key handle, return status code
UC	nCoreAPI Generate Logical Token	Input arguments	Logical token hash, key handle, return status code
UC	nCoreAPI Message digest	Input arguments, mechanism, data to be hashed	Hashed data, return status code
UC	nCoreAPI Modular Exponentiation	Input arguments	Operation result, return status code
UC	nCoreAPI Module hardware information	Input arguments	Hw info, return status code
UC	nCoreAPI No Operation	Input arguments	return status code
UC	nCoreAPI Change Share Passphrase	Input arguments, slot, old pin, new pin	return status code

Role	Service	Input	Output
NSO	nCoreAPI NVRAM Allocate	Input arguments, file info, acl	return status code
UC	nCoreAPI NVRAM Free	Input arguments, file name	return status code
UC	nCoreAPI Operation on NVM list	Input arguments	Files info, return status code
UC	nCoreAPI Operation on NVM files	Input arguments, file name, operation	Operation result, return status code
UC	nCoreAPI Key export	Input arguments, key handle	Wrapped key, return status code
UC	nCoreAPI Read file	Input arguments, file info, slot	File data, return status code
UC	nCoreAPI Read share	Input arguments, slot, key handle	return status code
UC	nCoreAPI Send share to remote slot	Input arguments	Encrypted share, return status code
UC	nCoreAPI Receive share from remote slot	Input arguments, encrypted share	return status code
UC	nCoreAPI Redeem Ticket	Input arguments, ticket	Key handle, return status code
UC	nCoreAPI Remote Administration	Input arguments, apdu payload	apdu payload, return status code
UC	nCoreAPI Destroy	Input arguments, key handle	return status code
UC	nCoreAPI Report statistics	Input arguments	Statistics, return status code
UC	nCoreAPI Show Status	Input arguments	Status and versioning info, return status code
UC	nCoreAPI Set ACL	Input arguments, key handle, acl	return status code
UC	nCoreAPI Set key application data	Input arguments, key handle, app data	return status code
NSO	nCoreAPI Set NSO Permissions	Input arguments, hash of KNSO (KNSO), permissions	return status code
UC	nCoreAPI Signature generation	Input arguments, mechanism, priv key handle, data to be signed	Signed data, return status code
UC	nCoreAPI Sign Module State	Input arguments	Certificate, return status code
UC	nCoreAPI Signature verification	Input arguments, mechanism, pub key handle, data to be verified	return status code
NSO	nCoreAPI Write file	Input arguments, file info, slot, data	return status code
UC	nCoreAPI Write share	Input arguments, file info, slot	Data, return status code

Table 8 Roles, Service Commands, Input and Output

Role	Authentication Method	Authentication Strength
PCO UC	ECDSA P-256, P-521 client key authentication as part of establishing an SSH based secure channel. Identity-based and required.	The ciphersuites used are: `ecdh-sha2-nistp256 , aes128-gcm@openssh.com, aes128-ctr@openssh.com, hmac-sha2-256-etm@openssh.com` 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.

Role	Authentication Method	Authentication Strength
		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.
NSO	smartcard authentication. Identity-based and required.	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 9 Roles and Authentication

4.2 Services

The following table describes the services provided by the Cryptographic Module and the access policy.

The Access column presents the access level given to the SSP

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

4.2.1 Setup service

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
info	This is a *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 options set with the setopt subcommand.	Data encryption/decryption (Cert. #A2512: AES CTR, GCM) MAC generation / verification (Cert. #A2512: HMAC)	KSESSION - SSH	PCO	E	return status code 0
factorystate	This is the *Perform Zeroisation* service. It zeroizes unprotected SSPs and returns the module to factory state. It then initiates a module reboot.	Data encryption/decryption (Cert. #A2512: AES CTR, GCM) MAC generation / verification (Cert. #A2512: HMAC)	KRESET (note: any SSP that is derived from KRESET, or protected by an SSP derived from KRESET, will also be effectively zeroised.) KSESSION - SSH	PCO	Z E G (note: SSH server authentication keys are only generated on first reboot after a	return status code 0

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
		Key generation (Cert. #A2513: ECDSA KeyGen, vendor affirmed: CKG)	KSSH_SETUP, KSSH_SSHADMIN, KSSH_MONITOR, KSSH_UPDATER, KRESET		`setup factorystate` command)	
settime	This subcommand sets the system date and time.	Data encryption/decryption (Cert. #A2512: AES CTR, GCM) MAC generation / verification (Cert. #A2512: HMAC)	KSESSION - SSH	PCO	E	return status code 0
gettime	This subcommand returns the system date and time.	Data encryption/decryption (Cert. #A2512: AES CTR, GCM) MAC generation / verification (Cert. #A2512: HMAC)	KSESSION - SSH	PCO	E	return status code 0

Table 10 Approved Services

The approved service indicator is the successful completion of these services (return status code 0), as they only use approved mechanisms.

4.2.2 SSHAdmin service

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
set	Loads the client public key in the module, that will be used to authenticate the requester of a particular service	Data encryption/decryption (Cert. #A2512: AES CTR, GCM) MAC generation / verification (Cert. #A2512: HMAC)	KSSH_CLIENT pub KSESSION - SSH	PCO	W E	return status code 0
list	Obtains the client public key for the service given by the 'role' parameter.	Data encryption/decryption (Cert. #A2512: AES CTR, GCM) MAC generation / verification (Cert. #A2512: HMAC)	KSSH_CLIENT pub KSESSION - SSH	PCO	R E	return status code 0
get-serverkey	Obtains the server public key for the service given by the 'role' parameter.	Data encryption/decryption (Cert. #A2512: AES CTR, GCM) MAC generation / verification (Cert. #A2512: HMAC)	KSSH_NCORE pub KSSH_UPDATER pub KSSH_SETUP pub KSSH_SSHADMIN pub KSSH_MONITOR pub KSESSION - SSH	PCO	R R R E	return status code 0

Table 11 Approved Services

The approved service indicator is the successful completion of these services (return status code 0), as they only use approved mechanisms.

4.2.3 Updater service

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
info	This is a *Show Module's Versioning Information* service. Obtains the version number of the HSM firmware.	Data encryption/decryption (Cert. #A2512: AES) MAC generation / verification (Cert. #A2512: HMAC)	KSESSION - SSH	PCO	E	return status code 0
receive	Transmits a file (intended to be an npkg upgrade file) to the HSM.	Data encryption/decryption (Cert. #A2512: AES) MAC generation / verification (Cert. #A2512: HMAC)	KSESSION - SSH	PCO	E	return status code 0
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.	Digital signature verification (Cert. #A2513: RSA SigVer, ECDSA SigVer) Data encryption/decryption (Cert. #A2512: AES) MAC generation / verification (Cert. #A2512: HMAC)	NSBIK pub NPSK pub NFIK pub NLIK pub KSESSION - SSH	PCO	E E E E E	return status code 0

Table 12 Approved Services

The approved service indicator is the succçessful completion of these services (return status code 0), as they only use approved mechanisms.

4.2.4 Monitor service

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
getlog	This is the *Show Status* service. Obtains the log of the system	Data encryption/decryption (Cert. #A2512: AES CTR, GCM) MAC generation / verification (Cert. #A2512: HMAC)	KSESSION - SSH	PCO	E	return status code 0
clearlog	Clears the log of the system	Data encryption/decryption (Cert. #A2512: AES CTR, GCM) MAC generation / verification (Cert. #A2512: HMAC)	KSESSION - SSH	PCO	E	return status code 0

Table 13 Approved Services

The approved service indicator is the successful completion of these services (return status code 0), as they only use approved mechanisms.

4.2.5 Discovery service

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
python-zeroconf	The (logical) network connection between host and HSM uses TCP/IP protocols over the PCIe bus; however, for ease of setup it needs to avoid requiring IP configuration by the user. Therefore the HSM uses zeroconf: each end of the virtual 'network segment' has only a link-local (IPv4 and IPv6) address. The HSM's address can be discovered by the host using multicast DNS, responding to mDNS queries.	-	-	Unauthenticated	-	return status code 0

Table 14 Approved Services

The approved service indicator is the successful completion of these services (return status code 0), as they only use approved mechanisms.

4.2.6 nCoreAPI service

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Big number operation Cmd_BignumOp	Performs an operation on a large integer.	-	-	UC	-	return status OK
Make Blob Cmd_MakeBlob	Creates a Key blob containing the key. Note that the key ACL needs to authorize the operation.	Key derivation (Cert. #A2513: KBKDF) Key wrapping (Cert. #A2513: AES CBC and HMAC, KTS-IFC)	KA, KRE_BLOBKEY, KR, KM, KNSO, LT BLOBKE, BLOBKM	UC	R E	return status OK
Bulk channel Cmd_ChannelOpen Cmd_ChannelUpdate	Provides a bulk processing channel for crypto operations	Encryption and decryption (Cert. #A2513: AES ECB, CBC, GCM) MAC generation and verification (Cert. #A2513: HMAC, KMAC, AES CMAC) Digital signature generation and verification (Cert. #A2513: RSA, ECDSA, DSA)	KA	UC	E	return status OK
Check User Action Cmd_CheckUserAction	Determines whether the ACL associated with a key allows a specific operator defined action.	-	KNSO, KA	UC	R	return status OK
Clear Unit Cmd_ClearUnit	This is the *Perform Self-tests* service. Zeroises all keys, tokens and shares that are loaded in RAM. Will cause the module	-	KA, KR, IMPATHKE, IMPATHKM, RAKME, RAKMA	UC	Z	return status OK

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
	to reboot and perform self-tests.					
Set Module Key Cmd_SetKM	Allows a key to be stored internally as a Module key (KM) value. The ACL needs to authorize this operation.	Message digest (Cert. #A2513: SHA-1)	KM	NSO	W	return status OK
Remove Module Key Cmd_RemoveKM	Deletes the KM with a given KM hash value from non-volatile memory.	-	KM	NSO	Z	return status OK
Duplicate key handle Cmd_Duplicate	Creates a second instance of a Key with the same ACL and returns a handle to the new instance. Note that the source key ACL needs to authorize this operation.	-	KA	UC	R	return status OK
Enable feature Cmd_StaticFeatureEnable	Enables the service. This service requires a certificate signed by the Master Feature Enable key.	-	-	UC	-	return status OK
Encryption Cmd_Encrypt	Encryption using the provided key handle.	Data encryption (Cert. #A2513: AES ECB, CBC, GCM)	KA	UC	E	return status OK
Decryption Cmd_Decrypt	Decryption using the provided key handle.	Data decryption (Cert. #A2513: AES ECB, CBC, GCM)	KA	UC	E	return status OK
Erase from smartcard /softcard Cmd_EraseFile Cmd_EraseShare	Removes a file or a share from a smartcard or softcard	-	-	UC	-	return status OK
Format Token Cmd_FormatToken	Formats a smartcard or a softcard.	-	-	UC	-	return status OK
File operations Cmd_FileCopy Cmd_FileCreate Cmd_FileErase Cmd_FileOp	Performs file operations in the module.	-	-	UC	-	return status OK
Force module to fail Cmd_Fail	Causes the module to enter a failure state.	-	-	UC	-	return status OK
Generate prime number Cmd_GeneratePrime	Generates a random prime.	Random bit generation (Cert. #A2513: DRBG)	DRBG entropy input, seed, internal state ('V' and 'C')	UC	E	return status OK
Generate random number	Generates a random number from the Approved DRBG.	Random bit generation (Cert. #A2513: DRBG)	DRBG entropy input, seed,	UC	E	return status OK

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Cmd_GenerateRandom			internal state ('V' and 'C')			
Get ACL Cmd_GetACL	Get the ACL of a given key.	-	KA	UC	R	return status OK
Get key application data Cmd_GetAppData	Get the application data field from a key.	-	KA	UC	R	return status OK
Get challenge Cmd_GetChallenge	Get a random challenge that can be used in fresh certificates.	Random bit generation (Cert. #A2513: DRBG)	DRBG entropy input, seed, internal state ('V' and 'C')	UC	E	return status OK
Get KLF2 Cmd_GetKLF2	Get a handle to the Module Long Term (KLF2) public key.	-	-	UC	-	return status OK
Get Key Information Cmd_GetKeyInfo Cmd_GetKeyInfoEx	Get the type, length and hash of a key.	Message digest (Cert. #A2513: SHA-1)	KA	UC	R	return status OK
Get module signing key Cmd_GetKML	Get a handle to the KML public key.	-	KML	UC	R	return status OK
Get list of slot in the module Cmd_GetSlotList	Get the list of slots that are available from the module.	-	-	UC	-	return status OK
Get Logical Token Info Cmd_GetLogicalTokenInfo Cmd_GetLogicalTokenInfoEx	Get information about a Logical Token: hash, state and number of shares.	Message digest (Cert. #A2513: SHA-1)	LT	UC	R	return status OK
Get list of module keys Cmd_GetKMList	Get the list of the hashes of all module keys and the KNSO.	Message digest (Cert. #A2513: SHA-1)	KM, KNSO, HKNSO	UC	R	return status OK
Get module state Cmd_GetModuleState	Returns unsigned data about the current state of the module.	-	-	UC	-	return status OK
Get real time clock Cmd_GetRTC	Get the current time from the module Real Time Clock.	-	-	UC	-	return status OK
Get share access control list Cmd_GetShareACL	Get the Share's ACL.	-	SHAREKEY	UC	R	return status OK
Get Slot Information Cmd_GetSlotInfo	Get information about shares and files on a Smartcard that has been inserted in a module slot.	-	-	UC	-	return status OK
Get Ticket Cmd_GetTicket	Get a ticket (an invariant identifier) for a key. This can be passed to another client or to a SEE World which can redeem it using Redeem	-	-	UC	-	return status OK

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
	Ticket to obtain a new handle to the object.					
Initialize Unit Cmd_InitializeUnit Cmd_InitializeUnitEx	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.	Key generation (Cert. #A2513: CKG, RSA, DSA) Message digest (Cert. #A2513: SHA-1)	KA, KRE_BLOBKEY, KR, KM, KAL, KML, KNSO, HKNSO, LT	UC	Z, G	return status OK
Insert a Softcard Cmd_InsertSoftToken	Allocates memory on the module that is used to store the logical token share and other data objects.	-	-	UC	-	return status OK
Remove a Softcard Cmd_RemoveSoftToken	Removes a Softcard from the module. It returns the updated shares and deletes them from the module's memory.	-	-	UC	-	return status OK
Impath channel Cmd_ImpathGetInfo Cmd_ImpathKXBegin Cmd_ImpathKXFinish Cmd_ImpathReceive Cmd_ImpathSend	Support for Impath channel. Requires Feature Enabled.	Key agreement (Cert. #A2513: KAS-FFC, Safe Primes Generation, Safe Primes Verification) Key derivation (Cert. #A2513: KBKDF) Data encryption and decryption (Cert. #A2513: AES CBC, GCM) MAC generation and verification (Cert. #A2513: HMAC-SHA256)	KML, IMPATHKE, IMPATHKM	UC	G, E	return status OK
Key generation Cmd_GenerateKey Cmd_GenerateKeyPair	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 certificate with the hash of the key and its ACL information.	Key generation (Cert. #A2513: CKG, RSA, ECDSA, DSA, KAS-ECC, KAS-FFC, Safe Primes Generation) Digital signature generation (Cert. #A2513: DSA)	KML, KA, DRBG entropy input, seed, internal state ('V' and 'C')	UC	G	return status OK
Key import Cmd_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.	-	KA	UC	W	return status OK
Derive Key Cmd_DeriveKey	Performs key wrapping, unwrapping, transport, exchange and derivation. The ACL needs to authorize this operation.	Key derivation (Cert. #A2513: KBKDF)	KA	UC	R, W	return status OK

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
		Key wrapping/unwrapping (Cert. #A2513: KTS-AES) Key transport (Cert. #A2513: KTS-IFC) Key agreement (Cert. #A2513: KAS-FFC, KAS-ECC)				
Load Blob Cmd_LoadBlob	Load a Key blob into the module. It returns a handle to the key suitable for use with module services.	Key derivation (Cert. #A2513: KBKDF) Key unwrapping (Cert. #A2513: AES CBC and HMAC, KTS-IFC)	KA, KRE_BLOBKEY, KR, KM, KNSO BLOBKE, BLOBKM	UC	W E	return status OK
Load Logical Token Cmd_LoadLogicalToken	Initiates loading a Logical Token from Shares, which can be loaded with the Read Share command.	-	-	UC	-	return status OK
Generate Logical Token Cmd_GenerateLogicalToken	Creates a new Logical Token with given properties and secret sharing parameters.	Key generation (Cert. #A2513: CKG)	KM, LT	UC	G, W	return status OK
Message digest Cmd_Hash	Computes the cryptographic hash of a given message.	Message digest (Cert. #A2513: SHS)	-	UC	-	return status OK
Modular Exponentiation Cmd_ModExp Cmd_ModExpCrt Cmd_RSAImmedVerifyEncrypt Cmd_RSAImmedSignDecrypt	Performs a modular exponentiation (standard or CRT) on values supplied with the command.	-	-	UC	-	return status OK
Module hardware information Cmd_ModuleInfo	Reports detailed hardware information.	-	-	UC	-	return status OK
No Operation Cmd_NoOp	No operation.	-	-	UC	-	return status OK
Change Share Passphrase Cmd_ChangeSharePIN Cmd_ChangeShareGroupPIN	Updates the passphrase of a Share.	Key derivation (KBKDF) Key wrapping/unwrapping (Cert. #A2513: AES CBC and HMAC, KTS-IFC)	SHAREKEY, LT, KM	UC	G, E, R, W	return status OK
NVRAM Allocate Cmd_NVMemAllocate	Allocation in NVRAM.	-	-	NSO	-	return status OK
NVRAM Free Cmd_NVMemFree	Deallocation from NVRAM.	-	-	UC	-	return status OK

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Operation on NVM list Cmd_NVMemList	Returns a list of files in NVRAM.	-	-	UC	-	return status OK
Operation on NVM files Cmd_NVMemOp	Operation on an NVRAM file.	-	-	UC		return status OK
Key export Cmd_Export	Exports a key in plain text. Note: in approved mode, only public keys can be exported.	-	KA	UC	R	return status OK
Read file Cmd_ReadFile	Reads data from a file on a Smartcard or Softcard. The ACL needs to authorize this operation.	-	-	UC	-	return status OK
Read share Cmd_ReadShare	Reads a share from a Smartcard or Softcard. Once a quorum of shares have been loaded, the module re-assembles the Logical Token.	Key derivation (Cert. #A2513: KBKDF) Key unwrapping (Cert. #A2513: AES CBC and HMAC)	SHAREKEY, LT, KM	UC	G, E, R	return status OK
Send share to remote slot Cmd_SendShare	Reads a Share and encrypts it with the Impath session keys for transmission to the peer module.	Data encryption(Cert. #A2513: AES CBC, GCM) MAC generation (Cert. #A2513: HMAC-SHA256)	IMPATHKE, IMPATHKM, SHAREKEY	UC	R, E	return status OK
Receive share from remote slot Cmd_ReceiveShare	Receives a Share encrypted with the Impath session keys by a remote module.	Data decryption (Cert. #A2513: AES CBC, GCM) MAC verification (Cert. #A2513: HMAC-SHA256)	IMPATHKE, IMPATHKM, SHAREKEY	UC	R, E	return status OK
Redeem Ticket Cmd_RedeemTicket	Gets a handle in the current name space for the object referred to by a ticket created by Get Ticket.	-	-	UC	-	return status OK
Remote Administration Cmd_DynamicSlotCreateAssociation Cmd_DynamicSlotExchangeAPDUs Cmd_DynamicSlotsConfigure Cmd_DynamicSlotsConfigureQuery Cmd_VerifyCertificate	Provides remote presentation of Smartcards using a secure channel between the module and the Smartcard.	Key agreement (Cert. #A2513: KAS-ECC ECDH) Key derivation (Cert. #A2513: KBKDF) Data encryption and decryption (Cert. #A2513: AES CBC) MAC generation and verification (Cert. #A2513: AES CMAC) Digital signature verification (Cert. #A2513: ECDSA)	RAKME, RAKMA, KWARN_pub	UC	G, E E	return status OK
Destroy Cmd_Destroy	Remove handle to an object in RAM. If the current handle is	-	KA, KNSO, LT	UC	Z	return status OK

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
	the only one remaining, the object is zeroised from RAM.					
Report statistics Cmd StatGetValues Cmd_StatEnumTree	Reports the values of the statistics tree.	-	-	UC	-	return status OK
Show Status Cmd_NewEnquiry	This is a *Show Status and Show Module's Versioning Information* service. Report status information.	-	-	UC	-	return status OK
Set ACL Cmd_SetACL	Replaces the ACL of a given key with a new ACL. The ACL needs to authorize this operation.	-	KA	UC	W	return status OK
Set key application data Cmd_SetAppData	Writes the application information field of a key.	-	KA	UC	W	return status OK
Set NSO Permissions Cmd_SetNSOPerms	Sets the NSO key hash and which permissions require a Delegation Certificate.	-	HKNSO	NSO	W	return status OK
Signature generation Cmd_Sign	Generate a digital signature or MAC value.	MAC generation (Cert. #A2513: HMAC, KMAC, AES CMAC) Digital signature generation (Cert. #A2513: RSA, ECDSA, DSA)	KA, KNSO	UC	E	return status OK
Sign Module State Cmd_SignModuleState	Returns a signed certificate that contains data about the current configuration of the module.	Digital signature generation (Cert. #A2513: DSA)	KML	UC	E	return status OK
Signature verification Cmd_Verify	Verifies a digital signature or MAC value.	MAC verification (Cert. #A2513: HMAC, KMAC, AES CMAC) Digital signature verification (Cert. #A2513: RSA, ECDSA, DSA)	KA	UC	E	return status OK
Write file Cmd_WriteFile	Writes a file to a Smartcard or Softcard.	-	-	NSO	-	return status OK
Write share Cmd_WriteShare	Writes a Share to a Smartcard or Softcard.	Key derivation (Cert. #A2513: KBKDF) Key wrapping (Cert. #A2513: AES CBC and HMAC)	SHAREKEY, LT, KM	UC	G, E, W	return status OK

Table 15 Approved Services

Non-approved services will fail with a return an error code indicator "StrictFIPS140".

All nCore API services are sent through the SSH channel, performing security functions Data encryption/decryption, MAC generation / verification and access E (execute) of the session keys KSESSION - SSH.

5 Software/Firmware security

The nShield 5s cryptographic module's executable code is delivered by Entrust as a single signed firmware package (.npkg file). The bootloader and firmware integrity is verified at start up using RSA with 4096 bit key and SHA2-256. The Library Partition is an internal storage area that contains a number of auxiliary files required for operation of the module. The integrity of the Library Partition is verified using ECDSA with curve P-521 and SHA2-512.

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

6 Operational environment

Not applicable. The module has a limited operational environment, it is designed to accept only controlled firmware changes that successfully pass the software/firmware load test.

7 Physical security

The product is a multi-chip embedded Cryptographic Module, as defined in FIPS 140-3. It is enclosed in a hard and opaque epoxy resin which meets the physical security requirements of FIPS 140-3 level 3.

The cryptographic module implements Environmental Failure Protections (EFP) which detect out of range voltage and temperature and shuts down the module.

Physical Security Mechanism	Recommended Frequency of Inspection/Test	Inspection/Test Guidance Details
Hard and opaque epoxy	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 16 Physical Security Inspection Guidelines

	Temperature or voltage measurement	Specify EFP or EFT	Specify if this condition results in a shutdown or zeroisation
Low Temperature	0ºC	EFP	Shutdown
High Temperature	95ºC	EFP	Shutdown
Low Voltage	2.463V (battery) 0V (PCIe)	EFP	Shutdown
High Voltage	3.553V (battery) 14.45V (PCIe)	EFP	Shutdown

Table 17 EFP/EFT

	Hardness tested temperature measurement
Low Temperature	0ºC
High Temperature	95ºC

Table 18 Hardness testing temperature ranges

8 Non-invasive security

Not applicable.

9 Sensitive security parameters management

9.1 Keys and Sensitive Security Parameters

This section defines the Sensitive Security Parameters (SSPs) managed by the cryptographic module.

9.1.1 Platform SSPs

Key/SSP/Name/Type	Strength	Security Function and Cert. Number	Generation	Import/Export	Establishment	Storage	Zeroisation	Use & related keys
KRESET (CSP)	256 bits	Re-settable key A2513	DRBG	Never	n/a	MSP430 FRAM, in plaintext	factorystate	Key derivation
KUSER_SSH (CSP)	256 bits	Global SSH key encryption key AES-256 A2513	Derived at start-up using KBKDF from KRESET and other fixed parameters	Never	n/a	RAM, in plaintext	Power cycle factorystate	Encryption of all the service's server authentication SSH keys
KSSH_SETUP (CSP)	128 bits	Server authentication SSH key for the Setup service ECDSA P-256 A2513 A2512	DRBG	Private key: never Public key: output via SSHAdmin service	n/a	In Flash, encrypted with KUSER_SSH	factorystate	SSH channel session
KSSH_UPDATER (CSP)	128 bits	Server authentication SSH key for the Updater service ECDSA P-256 A2513 A2512	DRBG	Private key: never Public key: output via SSHAdmin service	n/a	In Flash, encrypted with KUSER_SSH	factorystate	SSH channel session
KSSH_SSHADMIN (CSP)	128 bits	Server authentication SSH key for the SSH Admin service ECDSA P-256 A2513 A2512	DRBG	Private key: never Public key: output via SSHAdmin service	n/a	In Flash, encrypted with KUSER_SSH	factorystate	SSH channel session
KSSH_MONITOR (CSP)	128 bits	Server authentication SSH key for the Monitor service	DRBG	Private key: never Public key: output via	n/a	In Flash, encrypted with KUSER_SSH	factorystate	SSH channel session

Key/SSP/Name/Type	Strength	Security Function and Cert. Number	Generation	Import/Export	Establishment	Storage	Zeroisation	Use & related keys
		ECDSA P-256 A2513 A2512		SSHAdmin service				
KSESSION - SSH (CSP)	128 bits	SSH channel session keys AES GCM or AES CTR, HMAC A2513	n/a	Never	ECDH	RAM, in plaintext	Power cycle or channel closure factorystate	SSH channel data encryption and integrity
NSBIK pub (not an SSP)	128 bits	Bootloader public integrity key RSA 4096 bit A2404 A6385	Entrust	Import: Firmware update Export: Never	n/a	Flash, in plaintext	n/a	Firmware integrity test
NFIK pub (not an SSP)	128 bits	Firmware public signature verification key RSA 4096 bit A2404 A6385	Entrust	Import: Firmware update Export: Never	n/a	Flash, in plaintext	n/a	Firmware integrity test
NLIK pub (not an SSP)	256 bits	Library public integrity key ECDSA P-521 A2513	Entrust	Import: Firmware update Export: Never	n/a	Flash, in plaintext	n/a	Firmware integrity test
NPSK pub (PSP)	256 bits	Package public signature verification key ECDSA P-521 A2513	Entrust	Import: Firmware update Export: Never	n/a	Flash, in plaintext	n/a	Firmware loading test
KSSH_CLIENT pub (PSP)	128 bits	Client authentication SSH key for each of the services (Updater, Setup, SSHAdmin, nCoreAPI) ECDSA P-256 A2513	Client side	Through SSHAdmin service	n/a	Flash, in plaintext	factorystate	SSH authentication credentials
DRBG entropy input (CSP)	> 256 bits	Platform DRBG A2513	520 bits from the approved Entropy Source.	Never	n/a	RAM, in plaintext	Power cycle factorystate	Random number generation

Key/SSP/Name/Type	Strength	Security Function and Cert. Number	Generation	Import/Export	Establishment	Storage	Zeroisation	Use & related keys
DRBG seed (CSP)	256 bits	Platform DRBG A2513	Generated as per SP 800-90Arev1 with 696 bits from the approved Entropy Source: 520 bits entropy input 176 bits random nonce	Never	n/a	RAM, in plaintext	Power cycle factorystate	Random number generation
DRBG internal state ('V' and 'C' values) (CSP)	256 bits	Platform DRBG A2513	Generated as per SP 800-90Arev1.	Never	n/a	RAM, in plaintext	Power cycle factorystate	Random number generation

Table 19 Platform SSP table

nCore API service SSPs

9.1.1.1 Service SSPs

The following SSPs are related to the nCore API service.

Key/SSP/Name/Type	Strength	Security Function and Cert. Number	Generation	Import/Export	Establishment	Storage	Zeroisation	Use & related keys
KCONTAINER (CSP)	256 bits	Master key for container A2513	Derived at start-up using KBKDF from KRESET, container-id and other fixed value.	Never	n/a	RAM, in plaintext	Power cycle factorystate	Key derivation
KCONTAINERSSH (CSP)	256 bits	Encryption key for KSSH_NCORE AES-256 A2513	Derived at start-up using KBKDF from KCONTAINER, and other fixed value.	Never	n/a	RAM, in plaintext	Power cycle factorystate	Encryption
KSSH_NCORE (CSP)	128 bits	Server authentication ssh key for the nCore API service ECDSA P-256 A2513 A2512	DRBG	Private key: Never Public key output via SSH Admin service	n/a	In Flash, encrypted with KCONTAINERSSH	factorystate	SSH channel session

Table 20 nCoreAPI SSP table

9.1.1.2 Security World SSPs

The following SSPs are related to the Security World in which the cryptographic module is enrolled into.

Key/SSP/Name/Type	Strength	Security Function and Cert. Number	Generation	Import/Export	Establishment	Storage	Zeroisation	Use & related keys
KRE_BLOBKEY (CSP)	128 bits	Recovery confidentiality key RSA 3072 bit A2513	DRBG	Import: From key blob, decrypted with LTRE Export: in key blob, encrypted with LTRE	n/a	RAM, in plaintext	Power cycle Cmd_Destroy factorystate	Key used to protect recovery keys (KR).
KR (CSP)	256 bits	Recovery key AES 256 A2513	DRBG	Import: From key blob, decrypted with KRE_BLOBKEY Export: in key blob, encrypted with KRE_BLOBKEY	n/a	RAM, in plaintext	Power cycle Cmd_Destroy factorystate	Key used to derive (using SP 800-108 KDF in counter mode) the keys Ke (AES 256-bit) and Km (HMAC-SHA256) that protect an archive copy of an application key.
IMPATHKE IMPATHKM (CSP)	256 bits	Session keys for impath channel A2513	n/a	Never	DH	RAM, in plaintext	Power cycle or channel closure factorystate	Encryption and decryption MAC generation and verification
KA (CSP)	≥ 112 bits	Application keys A2513	DRBG	Import: From key blob, decrypted with LTA or KR Export: in key blob, encrypted with LTA or KR	n/a	RAM, in plaintext	Power cycle Cmd_Destroy factorystate	Application keys used for general purpose cryptographic services: • Encryption and decryption • Digital signature generation and verification • MAC generation and verification • Key derivation, key agreement.

Key/SSP/Name/Type	Strength	Security Function and Cert. Number	Generation	Import/Export	Establishment	Storage	Zeroisation	Use & related keys
KM (CSP)	256 bits	Security World module key AES 256 A2513	DRBG	Import: From key blob, decrypted with LTM Export: in key blob, encrypted with LTM	n/a	Flash, in plaintext	factorystate or Initialize Unit	Key used for key derivation to protect logical tokens and associated module Key Blobs.
KML (CSP)	128 bits	Module Signing key DSA 3072 bit A2513	DRBG	Never	n/a	Flash, in plaintext	factorystate or Initialize Unit	Digital signature generation for key generation certificates and module state certificates.
KNSO (CSP)	128 bits	NSO key DSA 3072 bit A2513	DRBG	Import: From key blob, decrypted with LTNSO Export: in key blob, encrypted with LTNSO	n/a	RAM, in plaintext	Power cycle Cmd_Destroy factorystate	nShield Security Officer key used for NSO authorisation and Security World integrity
HKNSO (PSP)	160 bits	Hash of public KNSO A2513	n/a	Never	Security World creation	Flash, in plaintext	factorystate or Initialize Unit	nShield Security Officer key used for NSO authorisation and Security World integrity
BLOBKE BLOBKM (CSP)	256 bits	Key blob encryption and MAC key AES 256 HMAC-SHA256 A2513	n/a	Never	Derived from LTx	RAM, in plaintext	Power cycle factorystate	Key wrapping
LTx (CSP)	256 bits	Logical token for key x AES 256 A2513	DRBG	Import: From quorum of encrypted Shares using Shamir Secret Scheme Export: To encrypted Shares using Shamir Secret Scheme	From Shares using Shamir Secret Schee	RAM, in plaintext	Power cycle Cmd_Destroy factorystate	Key derivation
SHAREKEY (CSP)	256 bits	Share encryption and MAC keys AES 256 HMAC-SHA256 A2513	n/a	Never	Derived from KM and other additional data	RAM, in plaintext	Power cycle factorystate	Protects a share when written to a smartcard or softcard. This key is used to derive using KBKDF the keys Ke and Km used to wrap the share.

Key/SSP/Name/Type	Strength	Security Function and Cert. Number	Generation	Import/Export	Establishment	Storage	Zeroisation	Use & related keys
RAKME RAKMA (CSP)	256 bits	Session keys for remote admin channel AES 256 A2513	n/a	Never	ECDH	RAM, in plaintext	Power cycle or channel closure factorystate	Encryption and decryption MAC generation and verification
KAL (CSP)	128 bits	Audit logging key DSA 3072-bit A2513	DRBG	Never	n/a	Flash, in plaintext	factorystate or Initialize Unit factorystate	Digital signature generation of the audit trail.
KWARN pub (PSP)	256 bits	Entrust root warranting public key for Administrator Cards and Operator Cards ECDSA P-521 A2513	Entrust	Import: fw update Export: never	n/a	Flash, in plaintext , as part of the firmware image	n/a protected	Digital signature verification to authenticate remote cards.
DRBG entropy input (CSP)	256 bits	nCoreAPI DRBG A2513	520 bits from Platform DRBG	Never	n/a	RAM, in plaintext	Power cycle factorystate	Random number generation
DRBG seed (CSP)	256 bits	nCoreAPI DRBG A2513	Generated as per SP 800-90Arev1 with 696 bits from Platform DRBG: 520 bits entropy input 176 bits random nonce	Never	n/a	RAM, in plaintext	Power cycle factorystate	Random number generation
DRBG internal state ('V' and 'C' values) (CSP)	256 bits	nCoreAPI DRBG A2513	Generated as per SP 800-90Arev1.	Never	n/a	RAM, in plaintext	Power cycle factorystate	Random number generation

Table 21 Security World SSP table

As per IG 9.7.B, the zeroisation of SSPs is explicitly indicated by the successful return code of the `setup factorystate` command. Temporary SSPs are zeroised implicitly.

9.2 SSP zeroization methods

Zeroization of all unprotected SSPs keys occurs immediately when the module is reset to the factory state with the `setup factorystate` command.

9.3 Entropy sources

The cryptographic module has a hardware based true random number generator used to seed the DRBGs.

Entropy sources	**Minimum number of bits of entropy**	**Details**
nShield 5s Physical True Random Number Generator ESV Cert. #E38	0.89 bits per output bit	Hardware entropy source compliant with SP 800-90B. 520 output bits are used for the entropy input of the DRBG, this is more than 256 bits of min-entropy.

Table 22 Random Bit Generator Entropy Source Specification

10 Self tests

The Cryptographic Module performs pre-operational, conditional and periodic self-tests. It also supports pre-operational self-tests on demand by resetting the module.

In the event of a self-test failure, the module enters the error state. While in this state, the module does not process any commands, and will indicate the error on the status LED and the error log, which can be retrieved with the command `monitor getlog`.

10.1 Pre-operational self tests

10.1.1 Integrity tests

At start up, the following integrity tests are performed:

- The bootloader integrity is verified using RSA with 4096 bit key and SHA2-256 (Cert. #A2404, #A6385), using NSBIK public key.
- The firmware integrity is verified using RSA with 4096 bit key and SHA2-256 (Cert. #A2404, #A6385), using NFIK public key.
- The library partition integrity is verified using ECDSA with curve P-521 and SHA2-512 (Cert. #A2513), using NLIK public key.

10.2 Conditional self tests

10.2.1 Crypto self tests

The following cryptographic algorithm self tests (CASTs) are run before the first use of any cryptographic mechanism.

Algorithm	Description
Bootloader crypto (Cert. #A2404, #A6385)	
SHA2-256	Known Answer Test
RSA	Known Answer Test: verification RSASSA-PKCS-v1_5 with 4096 bit key and SHA2-256
nCore crypto (Cert. #A2513)	
AES ECB encrypt	Known Answer Test: encryption with 128, 192 and 256-bit keys
AES ECB decrypt	Known Answer Test: decryption with 128, 192 and 256-bit keys
AES CMAC	Known Answer Test: MAC generation/verification with 128-bit key
SHA-1	Known Answer Test: SHA-1, other sizes are tested along with KAT HMAC
SHA-3	Known Answer Test: SHA3-224, SHA3-256, SHA3-384, SHA3-512
HMAC with SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	Known Answer Test

Algorithm	Description
RSA	Known Answer Test: sign/verify RSASSA-PKCS-v1_5 (SHA2-224, SHA2-256, SHA2-384, SHA2-512) with 2048-bit key Known Answer Test: encrypt/decrypt RSA-OAEP with 2048-bit key
DSA	Known Answer Test: sign/verify with 2048-bit key and SHA2-224
ECDSA	Known Answer Test: sign/verify with curves P-224 and B-233
KAS-FFC	Known Answer Test: DH Shared Secret Computation with MODP-2048, MODP-3072
KAS-ECC	Known Answer Test: ECDH Shared Secret Computation with curves P-256 and B-233
One-step KDF	Known Answer Test: with SHA2-256 auxiliary function
Two-step KDF	Known Answer Test: with HMAC-SHA256 auxiliary function
KBKDF	Known Answer Test: Counter KDF with CMAC-AES256
DRBG	Known Answer Test: instantiate, reseed, generate as per SP 800-90Arev1 section 11.3
SSH crypto (Cert. #A2512)	
AES GCM encrypt	Known Answer Test: encryption with 128 bit key
AES GCM decrypt	Known Answer Test: decryption with 128 bit key
AES CTR encrypt	Known Answer Test: encryption with 128 bit key
AES CTR decrypt	Known Answer Test: decryption with 128 bit key
HMAC with SHA2-256 and SHA2-512	Known Answer Test
KAS-ECC	Known Answer Test: ECDH Shared Secret Computation with curve P-256
ECDSA	Known Answer Test: sign/verify with curves P-256 and P-521
SSH KDF	Known Answer Test: with SHA2-256

10.2.2 SP 800-90B health tests

At start up, the SP 800-90B Adaptive Proportion Test and Repetition Count Test are run on the output bits of the entropy source.

These tests are also run continuously during operation of the entropy source.

10.2.3 Pair-wise consistency tests

The module performs a pair-wise consistency test when RSA (Cert. #A2513), DSA (Cert. #A2513), ECDSA (Cert. #A2513), DH (Cert. #A2513) and ECDH (Cert. #A2513 and Cert. #A2512) keys are generated.

10.2.4 Firmware load test

Prior to updating the firmware, the cryptographic module validates the integrity and authenticity of the image update package.

The module performs the following actions before replacing the current image:

- Code signature verification with NPSK pub. The signature algorithm is ECDSA with SHA2-512 using the P-521 curve (Cert. #A2513).
- Verification that the Version Security Number (VSN) of the new image is not less than the VSN of the current image. This check is done for roll-back protection.

Note: A firmware image version loaded into this module that is not shown on the module certificate is out of the scope of this validation and requires a separate FIPS 140-3 validation.

10.3 Periodic self tests

The following self tests are run periodically every 24 hours:

- nCore crypto (Cert. #A2513) and SSH crypto (Cert. #A2512) self tests
- SP 800-90B health tests

As per IG 10.3E Resolution #3 c), the bootloader is designed exclusively to launch the main firmware for the module. In this architecture, as the module isn’t complete until the main firmware is launched (replacing the bootloader in executable memory), it is redundant that the bootloader itself implement a mechanism to run periodic tests.

11 Life-cycle assurance

This section provides specific FIPS-related guidance to Administrators and Operators. This guidance is aimed to complement the product user and installation guides which are delivered with the cryptographic module.

11.1 Delivery

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).

11.2 Cryptographic module identification

This section provides instructions to inspect the cryptographic module´s fw and hw version information and ensure they correspond with the FIPS 140-3 validated versions.

11.2.1 FW 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": "13.2.4-280-7f4f0c24",
            "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	13.2.4-280-7f4f0c24

Field	Expected value
recovery-version	13.2.4-280-7f4f0c24
uboot-version	1.1.0-1245-b9bedfa or 1.4.1-0-edb84d6e

11.2.2 HW 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"`

11.3 Approved mode of operation

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

1. When the cryptographic module is in factory state, it first needs to be initialized with the Entrust supplied utility `hsmadmin enroll`.
2. Create a FIPS 140-3 level 3 compliant Security World using the Entrust supplied utility `new-world`, 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.

11.4 End of life

Per FIPS 140-3 section 7.11.8, in the event that the module is no longer deployed or intended for further use, the Crypto Officer shall zeroize and destroy the module. The module shall be taken to an electronics recycling facility that offers (and assures) the physical destruction of e-waste.

12 Mitigation of other attacks

Not applicable.

Contact Us

Web site	https://www.entrust.com
Support	https://nshieldsupport.entrust.com
Email Support	nShield.support@entrust.com
Online documentation:	Available from the Support site listed above.

You can also contact our Support teams by telephone, using the following numbers:

Europe, Middle East, and Africa

United Kingdom:	+44 1223 622 444 One Station Square Cambridge CB1 2GA UK

Americas

Toll Free:	+1 833 425 1990
Fort Lauderdale:	+1 954 953 5229 Sawgrass Commerce Center – A Suite 130, 13800 NW 14 Street Sunrise FL 33323 USA

Asia Pacific

Australia:	+61 9126 9070 World Trade Centre Northbank Wharf Siddeley St Melbourne VIC 3005 Australia
Japan:	+81 50 3196 4994
Hong Kong:	+852 3008 4994 31/F, Hysan Place 500 Hennessy Road Causeway Bay Hong Kong

To get help with
Entrust nShield HSMs

nShield.support@entrust.com

nshieldsupport.entrust.com

ABOUT ENTRUST CORPORATION

Entrust keeps the world moving safely by enabling trusted identities, payments and data protection. Today more than ever, people demand seamless, secure experiences, whether they're crossing borders, making a purchase, accessing e-government services or logging into corporate networks. Entrust offers an unmatched breadth of digital security and credential issuance solutions at the very heart of all these interactions. With more than 2,500 colleagues, a network of global partners, and customers in over 150 countries, it's no wonder the world's most entrusted organizations trust us.

ENTRUST
SECURING A WORLD IN MOTION