ENTRUST

SECURING A WORLD IN MOTION

nShield5s HSM Security Target

Version: 115

Date: 2024-04-24 15:52:37

Contents

1 Introduction

1.1 ST Reference

Title	nShield5s HSM Security Target
Version	Current Page Version 115

1.2 TOE Reference

nShield5s Hardware Security Module v13.5.1.

1.3 TOE Overview

1.3.1 TOE type

The TOE is a Hardware Security Module (HSM) in PCIe board form factor.

1.3.2 TOE description

The nShield5s Hardware Security Module (HSM) is a general-purpose Cryptographic Module that comes in a PCI express board form factor protected by a tamper resistant enclosure. It performs encryption, digital signing, and key management on behalf of an extensive range of commercial and custom-built Client Applications including public key infrastructures (PKIs), identity management systems, application-level encryption and tokenization, and code signing.

Client Applications (always excluded from the TOE scope) can either be:

- local applications (physically residing within the same hardware appliance and communicating with the TOE via the PCIe interface), and/or
- local embedded applications (i.e. CodeSafe applications, residing within the TOE hardware boundary).

As a deployment option, the nShield5s HSM can either be installed in a generic host PC/Server, or it can be hosted inside the nShield5c, which is a network-attached appliance 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. In both cases, the host machine is out of scope of this evaluation.

The physical scope of the TOE is the following:

Type	Name	Identifier	Form factor	Delivery
Hardware	nShield5s (Model number NC5536E)	PCB assembly part number: PCA10005-01 PCB assembly revision: 03	PCIe board	Courier
	nShield5s ((Model number NC5536N) for nShield5c (Model number NH2096)	PCB assembly part number: PCA10005-01 PCB assembly revision: 03	PCIe board embedded in nShield5c appliance	Courier
Firmware	nShield5s firmware image	**Primary:** 13.5.1 **Recovery:** 13.5.0 **Bootloader**: 1.4.1	binary image file in ISO image or DVD	Courier or web download
Documentation	nShield5 Common Criteria Evaluated Configuration Guide	V29	pdf file	Web download

The logical boundary of the TOE comprises the main firmware located inside the PCIe board, with the exception of any CodeSafe Client Application residing within the hardware TOE boundary.

The Figure below provides an overview of the TOE (in blue) in its environment.

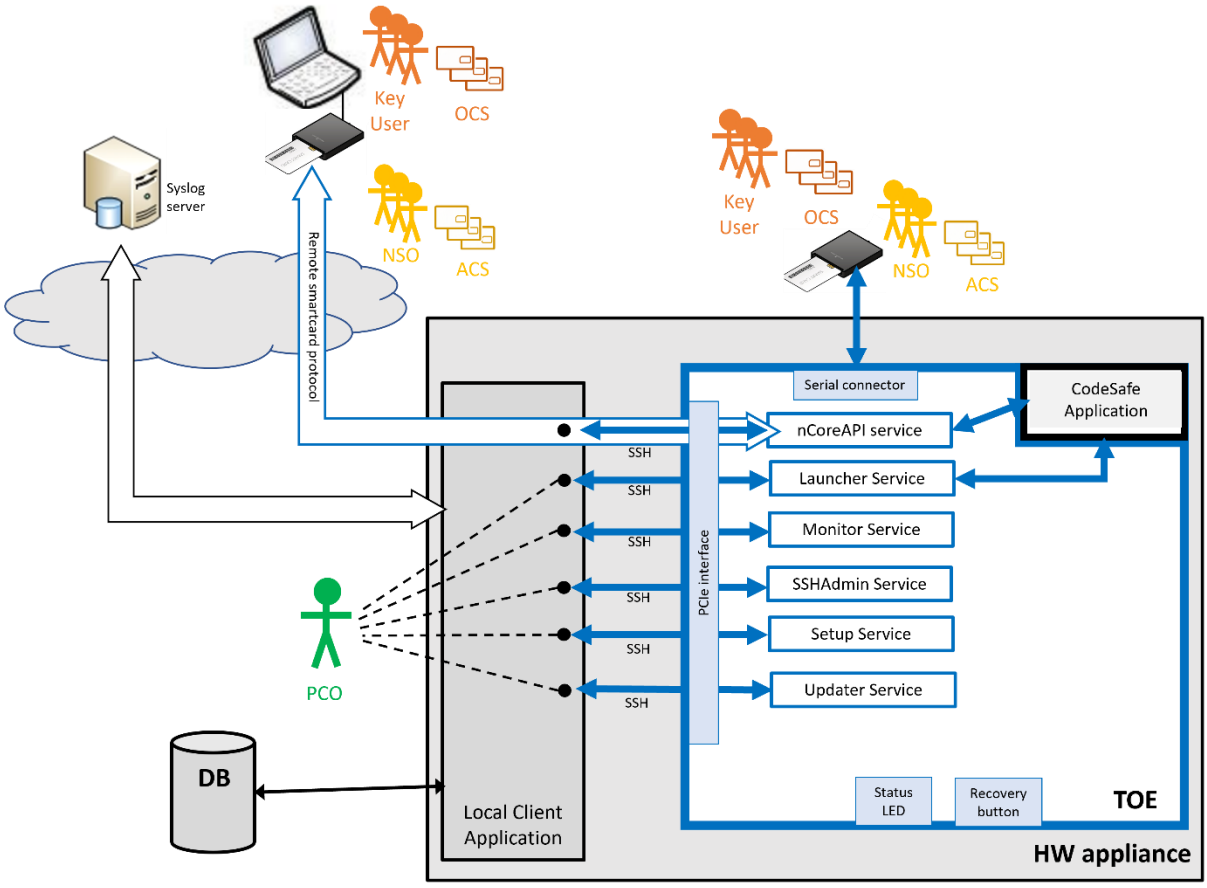

Figure 1 TOE in its environment.

1.3.2.1 TOE life-cycle

The following three life-cycle phases are identified:

- **TOE development phase**, covering design, production, test.
- **TOE delivery phase**, covering distribution to end users. Note that methods for delivery of TOE components to end users are specified in the table detailing the TOE physical scope.
- **TOE operational use phase**, covering TOE acceptance after delivery, identification, installation and usage.

1.3.3 Usage and major security features of the TOE

The TOE provides the following major security features:

- Cryptographic functions, including digital signature, encryption/decryption, key agreement, message digest, message authentication, key generation. See the tables below for a full overview of the CC-endorsed algorithms.
- Random Number Generation compliant with [AIS 31] and NIST [SP 800-90A],

- Secure key management,
- Secure logging of audit records,
- Physical tamper resistance meeting [ISO 19790] Level 3.

The TOE can be used as a general purpose Cryptographic Module in a wide range of use cases, including, but not limited to Trust Service Providers, for example with EN 419 241-2 to provide a QSCD for Remote Server Signing.

1.3.3.1 CC-endorsed algorithms

The following table describes the CC-endorsed cryptographic algorithms supported by the Cryptographic Module. The algorithms listed below are within the evaluation scope and are explicitly claimed in SFR FCS_COP.1.

Algorithm and mode	Standard	Key sizes	Use
Advanced Encryption Standard (AES) • ECB • CBC • GCM	FIPS 197 SP800-38A SP800-38D	128 bits 192 bits 256 bits	Data encryption/decryption
Advanced Encryption Standard (AES) • Key Wrapping (KW) • Key Wrapping with Padding (KWP) • GCM	SP800-38F SP800-38D	128 bits 192 bits 256 bits	Key wrapping/unwrapping (KTS)
Advanced Encryption Standard (AES) for SSH crypto • CTR • GCM	FIPS 197 SP800-38A SP800-38D	128 bits	Data encryption/decryption
RSA OAEP	SP 800-56Brev2	2048 bits 3072 bits 4096 bits	Key transport (KTS)
CKG	SP 800-133	128 bits	Symmetric key generation

Algorithm and mode	Standard	Key sizes	Use
		192 bits 256 bits	
RSA • RSASSA-PKCS-v1_5 • RSASSA-PSS	FIPS 186-4	1024 bits (verification only) 2048 bits 3072 bits 4096 bits	Key generation Signature generation and verification
Elliptic Curve Digital Signature Algorithm (ECDSA)	FIPS 186-4	• NIST P-224, P-256, P-384, P-521 • NIST K-233, K-283, K-409, K-571 • NIST B-233, B-283, B-409, B-571 • 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
Digital Signature Algorithm (DSA)	FIPS 186-4	L = 1024 bits, N = 160 bits (verification only) L = 2048 bits, N = 224 bits L = 2048 bits, N = 256 bits L = 3072 bits, N = 256 bits	Key generation Signature generation and verification
HMAC-SHA1 HMAC-SHA2	FIPS 198-1	>= 112 bits	MAC generation and verification
Advanced Encryption Standard (AES) • CMAC	SP800-38B	128 bits 192 bits 256 bits	MAC generation and verification
Diffie-Hellman (DH)	SP 800-56Arev3	MODP-2048 MODP-3072	Key establishment (KAS)

Algorithm and mode	Standard	Key sizes	Use
		MODP-4096 MODP-6144 MODP-8192 FB FC	
Elliptic Curve Diffie-Hellman (ECDH)	SP 800-56Arev3	• NIST P-224, P-256, P-384, P-521 • NIST K-233, K-283, K-409, K-571 • NIST B-233, B-283, B-409, B-571 • 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 establishment (KAS) and (KAS-SSC)
Elliptic Curve Menezes–Qu–Vanstone (ECMQV)	SP 800-56Arev3	• NIST P-224, P-256, P-384, P-521 • NIST K-233, K-283, K-409, K-571 • NIST B-233, B-283, B-409, B-571 • brainpoolP224r1/P224t1 (112 bits of strength) • brainpoolP256r1/P256t1 (128 bits of strength) • brainpoolP320r1/P320t1 (160 bits of strength) • brainpoolP384r1/P384t1 (192 bits of strength)	Key establishment (KAS)

Algorithm and mode	Standard	Key sizes	Use
		• brainpoolP512r1/P512t1 (256 bits of strength)	
Key Based KDF (KBKDF): • counter mode	SP 800-108	n/a	Key derivation
Secure Shell (SSH) KDF	SP 800-135rev1	n/a	Key derivation
SHA-1	FIPS 180-4	n/a	Message digest
SHA-224, SHA-256, SHA-384, SHA-512	FIPS 180-4	n/a	Message digest
SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 202	n/a	Message digest
Hash-based DRBG	SP 800-90A	344 bits	Random bit generation

Table 1 CC-endorsed Algorithms

1.3.3.2 Non-CC-endorsed algorithms

The TOE also provides other non-endorsed cryptographic services which shall not be used to meet CC compliance, as they are not claimed in FCS_COP.1 requirement of the Security Target.

It is responsibility of the user to ensure that the following alogorithms are not used, as the TOE does not prevent their usage in CMTs mode.

Algorithm/Function	Use
Symmetric	
DES	Data encryption/decryption Key wrapping/unwrapping
Triple DES encryption, MAC generation	
AES GCM with externally generated IV	
AES CBC MAC	
Aria	
Camellia	
Arc Four (compatible with RC4)	
CAST 256 (RFC2612)	

Algorithm/Function	Use
SEED (Korean Data Encryption Standard)	
Asymmetric	
KTS-OAEP-basic with SHA-256 with key size less than 2048 bits	Key transport Data encryption/decryption
ElGamal (encryption using Diffie-Hellman keys)	Key transport Data encryption/decryption
KCDSA (Korean Certificate-based Digital Signature Algorithm)	Signature generation and verification
RSA digital signature generation with SHA-1 or key size less than 2048 bits	
DSA digital signature generation with SHA-1 or key size less than 2048 bits	
ECDSA digital signature generation with SHA-1 or curves P-192, K-163 , B-163, brainpoolP160r1/P160t1, brainpoolP192r1/P192t1, other custom curves.	
Ed25519 public-key signature	
Deterministic DSA compliant with RFC6979	
DH with key size p < 2048 bits or q < 224 bits, or non-compliant with SP800-56Arev3	Key establishment
ECDH with curves P-192, K-163, B-163, brainpoolP160r1/P160t1, brainpoolP192r1/P192t1, other custom curves, or non-compliant with SP800-56Arev3	
EC MQV with curves P-192, K-163 or B-163, brainpoolP160r1/P160t1, brainpoolP192r1/P192t1, other custom curves., or non-compliant with SP800-56Arev3	
X25519 key exchange	
ECKA-EG key agreement	
ECIES encryption/wrapping and decryption/unwrapping	Key wrapping/unwrapping Data encryption/decryption
Hash	
HAS-160	Message digest
MD5	
RIPEMD-160	

Algorithm/Function	Use
Tiger	
Message Authentication Codes	
MD5, RIPEMD-160 and Tiger	MAC generation/verification
HMAC with key size less than 112 bits	
Other	
EMV support: Cryptogram (ARQC) generation and verification (includes EMV2000, M/Chip 4 and Visa Cryptogram Version 14, EMV 2004, M/Chip 2.1, Visa Cryptogram Version 10) Watchword generation and verification	MAC generation/verification
Hyperledger client side KDF	Key derivation
3GPP TUAK algorithm from ETSI standard [TUAK]	Key derivation
3GPP Milenage algorithm from ETSI standard [Milenage]	Key derivation

Table 2 Non-CC-endorsed Algorithms. Not Allowed in the CC-Mode of Operation

1.3.4 Non-TOE hardware/software/firmware required by the TOE

The following hardware and software, which is not part of the evaluation scope, is required by the TOE:

- nShield5c network appliance or host PC/Server, depending on the configuration,
- Host side software, including drivers, Hardserver, and high level APIs, which are provided by Entrust,
- Set of smartcards, TVD and module warrant certificate, provided by Entrust, for card-based authorisation,
- Syslog server for long-term storage of audit logs.

2 Conformance Claim

2.1 CC conformance claim

This Security Target is conformant to Common Criteria version 3.1 revision 5.

More precisely, this Security Target is:

- CC Part 2 extended [CC2],
- CC Part 3 conformant [CC3].

The assurance level of this Security Target is EAL4 augmented and extended by:

- AVA_VAN.5 Advanced methodical vulnerability analysis
- ALC_FLR.2 Flaw reporting procedures

2.2 PP claim

This Security Target claims strict conformance to Protection profiles for TSP Cryptographic modules - Part 5 Cryptographic Module for Trust Services version 1.0, [CEN EN 419221-5].

2.3 Conformance claim rationale

The Security Target includes all the Threats, OSPs, Assumptions and Objectives in the Protection Profile [CEN EN 419221-5].

The following SFRs, which are marked as optional in the Protection Profile, have been removed in this Security Target:

- FTP_TRP.1/External

3 Security Problem Definition

3.1 Assets

The assets that need to be protected by the TOE are identified below.

R.SecretKey: secret keys used in symmetric cryptographic functions and private keys used in asymmetric cryptographic functions, managed and used by the TOE in support of the cryptographic services that it offers. This includes user keys, owned and used by specific users, and support keys used in the implementation and operation of the TOE. The asset also includes copies of such keys made for external storage and/or backup purposes. The confidentiality and integrity of these keys must be protected.

R.PubKey: public keys managed and used by the TOE in support of the cryptographic services that it offers (including user keys and support keys). This asset includes copies of keys made for external storage and/or backup purposes. The integrity of these keys must be protected.

R.ClientData: data supplied by a client for use in a cryptographic function. Depending on the context, this data may require confidentiality and/or integrity protection.

R.RAD: reference data held by the TOE that is used to authenticate an administrator (hence to control access to privileged administrator functions such as TOE backup, export of audit data) or to authorise a user for access to secret and private keys (R.SecretKey). This asset includes copies of authentication/authorisation data made for external storage and/or backup purposes. The integrity of the RAD must be protected; its confidentiality must also be protected unless the authentication method used means that the RAD is public data (such as a public key).

3.2 Subjects

The types of subjects identified in this ST are:

S.Application: a client application, or process acting on behalf of a client application and that communicates with the TOE over a local or external interface. Client applications will in some situations be acting directly on behalf of end users (see S.User).

S.User: an end user of the TOE who can be associated with secret keys and authentication/authorisation data held by the TOE. An end user communicates with the TOE by using a client application (S.Application).

S.Admin: an administrator of the TOE. Administrators are responsible for performing the TOE initialisation, TOE configuration and other TOE administrative functions.

3.3 Threats

The following threats are defined for the TOE.

T.KeyDisclose Unauthorised disclosure of secret/private key

An attacker obtains unauthorised access to the plaintext form of a secret key (R.SecretKey), enabling either direct reading of the key or other copying into a form that can be used by the attacker as though the key were their own. This access may be gained during generation, storage, import/export, use of the key, or backup ~~if supported by the TOE~~.

T.KeyDerive Derivation of secret/private key

An attacker derives a secret key (R.SecretKey) from publicly known data, such as the corresponding public key or results of cryptographic functions using the key or any other data that is generally available outside the TOE.

T.KeyMod Unauthorised modification of a key

An attacker makes an unauthorised modification to a secret or public key (R.SecretKey or R.PubKey) while it is stored in, or under the control of, the TOE, including export and backups ~~if supported~~. This includes replacement of a key as well as making changes to the value of a key, or changing its attributes such as required authorisation, usage constraints or identifier (changing the identifier to the identifier used for another key would allow unauthorised substitution of the original key with a key known to the attacker). The threat therefore includes the case where an attacker is able to break the binding between a key and its critical attributes.

T.KeyMisuse Misuse of a key

An attacker uses the TOE to make unauthorised use of a secret key (R.SecretKey) that is managed by the TOE (including the unauthorised use of a secret key for a cryptographic function that is not permitted for that key), without necessarily obtaining access to the value of the key.

T.KeyOveruse Overuse of a key

An attacker uses a key (R.SecretKey) that has been authorised for a specific use (e.g. to make a single signature) in other cryptographic functions that have not been authorised.

T.DataDisclose Disclosure of sensitive client application data

An attacker gains access to data that requires protection of confidentiality (R.ClientData, and possibly R.RAD) supplied by a client application during transmission to or from the TOE or during transmission between physically separate parts of the TOE.

T.DataMod Unauthorised modification of client application data

An attacker modifies data (R.ClientData such as DTBS/R, authentication/authorisation data, or a public key (R.PubKey)) supplied by a client application during transmission to the TOE ~~or during transmission between physically separate parts of the TOE~~, so that the result returned by the TOE (such

as a signature or public key certificate) does not match the data intended by the originator of the request.

T.Malfunction Malfunction of TOE hardware or software

The TOE may develop a fault that causes some other security property to be weakened or to fail. This may affect any of the assets and could result in any of the other threats being realised. Particular causes of faults to be considered are:

- Environmental conditions (including temperature and power)
- Failures of critical TOE hardware components (including the RNG)
- Corruption of TOE software

3.4 Organisational Security Policies

P.Algorithms Use of approved cryptographic algorithms

The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognised authorities as appropriate for use by TSPs.

P.KeyControl Support for control of keys

The life cycle of the TOE and any secret keys that it manages (where such keys are associated with specific entities, such as the signature creation data associated with a signatory or the seal creation data associated with a seal creator), shall be implemented in such a way that the secret keys can be reliably protected by the legitimate owner against use by others, and in such a way that the use of the secret keys by the TOE can be confined to a set of authorised cryptographic functions.

P.RNG Random Number Generation

The TOE is required to generate random numbers that meet a specified quality metric, for use by client applications. These random numbers shall be suitable for use as keys, authentication/authorisation data, or seed data for another random number generator that is used for these purposes.

P.Audit Audit trail generation

The TOE is required to generate an audit trail of security-relevant events, recording the event details and the subject associated with the event.

Application note: The TOE is assumed to be part of a larger system that manages audit data. The TOE therefore logs audit records, and it is assumed that these are collected, maintained and reviewed in the larger system. Hence there is no separate auditor role within the cryptographic module TOE, but the role of System Auditor is assumed to exist in the larger system – cf. A.AuditSupport.

3.5 Assumptions

A.ExternalData Protection of data outside TOE control

Where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities must provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment.

In particular, any backups of the TOE and its data are maintained in a way that ensures appropriate controls over making backups, storing backup data, and using backup data to restore an operational TOE. The number of sets of backup data does not exceed the minimum needed to ensure continuity of the TSP service. The ability to restore a TOE to an operational state from backup data requires at least dual person control (i.e. the participation and approval of more than one authenticated administrator).

A.Env Protected operating environment

The TOE operates in a protected environment that limits physical access to the TOE to authorised Administrators. The TOE software and hardware environment (including client applications) is installed maintained by Administrators in a secure state that mitigates against the specific risks applicable to the deployment environment.

A.DataContext Appropriate use of TOE functions

Any client application using the cryptographic functions of the TOE will ensure that the correct data are supplied in a secure manner (including any relevant requirements for authenticity, integrity and confidentiality). For example, when creating a digital signature over a DTBS the client application will ensure that the correct (authentic, unmodified) DTBS/R is supplied to the TOE, and will correctly and securely manage the signature received from the TOE; and when certifying a public key the client application will ensure that necessary checks are made to prove possession of the corresponding private key. The client application may make use of appropriate secure channels provided by the TOE to support these security requirements. Where required by the risks in the operational environment a suitable entity (possibly the client application) performs a check of the signature returned from the TOE, to confirm that it relates to the correct DTBS.

Client applications are also responsible for any required logging of the uses made of the TOE services, such as signing (or sealing) events.

Similar requirements apply in local use cases where no client application need be involved, but in which the TOE and its user data (such as keys used for signatures) need to be configured in ways that will support the need for security requirements such as sole control of signing keys.

Appropriate procedures are defined for the initial creation of data and continuing operation of the TOE according to the specific risks applicable to the deployment environment and the ways in which the TOE is used.

A.UAuth Authentication of application users

Any client application using the cryptographic services of the TOE will correctly and securely gather identification and authentication/authorisation data from its users and securely transfer it to the TOE (protecting the confidentiality of the

authentication/authorisation data as required) when required to authorise the use of TOE assets and services.

A.AuditSupport Audit data review

The audit trail generated by the TOE will be collected, maintained and reviewed by a System Auditor according to a defined audit procedure for the TSP.

A.AppSupport Application security support

Procedures to ensure the ongoing security of client applications and their data will be defined and followed in the environment, and reflected in use of the appropriate TOE cryptographic functions and parameters, and appropriate management and administration actions on the TOE. This includes, for example, any relevant policies on algorithms, key generation methods, key lengths, key access, key import/export, key usage limitations, key activation, cryptoperiods and key renewal, and key/certificate revocation.

4 Security Objectives

This section identifies and defines the security objectives for the TOE and its operational environment.

Security objectives reflect the stated intent and counter the identified threats, as well as comply with the identified organisational security policies and assumptions.

4.1 Security Objectives for the TOE

The following security objectives describe security functions to be provided by the TOE.

OT.PlainKeyConf Protection of confidentiality of plaintext secret keys

The plaintext value of secret keys is not made available outside the TOE (except where the key has been exported securely in the manner of OT.ImportExport). This includes protection of the keys during generation, storage (including external storage), and use in cryptographic functions, and means that even authorised users of the keys and administrators of the TOE cannot directly access the plaintext value of a secret key.

OT.Algorithms Use of approved cryptographic algorithms

The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognised authorities as appropriate for use by TSPs. This ensures that the algorithms used do not enable publicly known data to be used to derive secret keys.

OT.KeyIntegrity Protection of integrity of keys

The value and critical attributes of keys (secret or public) have their integrity protected by the TOE against unauthorised modification (unauthorised modifications include making unauthorised copies of a key such that the attributes of the copy can be changed without the same authorisation as for the original key). Critical attributes in this context are defined to be those implementation-level attributes of a key that could be used by an attacker to cause the equivalent of a modification to the key value by other means (e.g. including changing the cryptographic functions for which a key can be used, the users with access to the key, or the identifier of the key). This objective includes protection of the keys during generation, storage (including external storage), and use.

OT.Auth Authorisation for use of TOE functions and data

The TOE carries out an authentication/authorisation check on all subjects before allowing them to use the TOE. The following types of entity are distinguished for the purposes of authorisation (i.e. each type has a distinct method of authorisation):

- administrators of the TOE,

- ~~users of TOE cryptographic functions (client applications using secure channels)~~,
- users of secret keys.

In particular, the TOE always requires authorisation before using a secret key.

Application Note: Local client applications within a suitable security environment (such as client applications that are connected to the TOE by a channel such as a PCIe bus within the same hardware appliance) do not require authentication to communicate with the TOE, as noted in section 1.3.1 of the PP. However, use of a secret key always requires prior authorisation.

OT.KeyUseConstraint Constraints on use of keys

Any key (secret or public) has an unambiguous definition of the purposes for which it can be used, in terms of the cryptographic functions or operations (e.g. encryption or signature) that it is permitted to be used for. The TOE rejects any attempt to use the key for a purpose that is not permitted. The TOE also has an unambiguous definition of the subjects that are permitted to access the key (and the purposes for which this access can be used) and allows this to be set to the granularity of an individual subject – these access constraints apply to use of the key even where the key value is not accessible. This objective means that the TOE also prevents unauthorised use of any cryptographic functions that use a key.

OT.KeyUseScope Defined scope for use of a key after authorisation

The TOE is required to define and apply clearly stated limits on when authorisation and reauthorisation are required in order for a secret key to be used. For example the TOE may allow secret keys to be used for a specified time period or number of uses after initial authorisation, or for may allow the key to be used until authorisation is explicitly rescinded. As another example, the TOE may implement a policy that requires re-authorisation before every use of a secret key.

Application Note: Such limits on the use of a key after initial authorisation are termed "re-authorisation conditions" in this PP. A wide range of policies and re-authorisation conditions are allowed, and different policies may be applied to different types of secret key, but the re-authorisation conditions for all types of secret key must be unambiguously defined in the Security Target. The decision to use supported reauthentication conditions is made on the basis of the application context. Making appropriate use of re-authorisation conditions supports client applications in meeting their requirements for OE.DataContext and OE.AppSupport.

OT.DataConf Protection of confidentiality of sensitive client application data

The TOE provides secure channels to client applications that can be used to protect the confidentiality of sensitive data (such as authentication/authorisation data) during transmission between the client application and the TOE~~, or during transmission between separate parts of the TOE where that transmission passes through an insecure environment~~.

Application Note: Protection of secret keys (as a specific type of sensitive data) is also subject to additional protection specified in other TOE objectives. Any requirements for secure storage and control of access to other types of client

application data within the TOE rely on the client application using appropriate interfaces and cryptographic functions to protect it, as required by OE.DataContext and OE.AppSupport. For example, if a client application uses the TOE to perform cryptographic functions on data that represent a passphrase value and the passphrase value is to be stored on the TOE, then the client application would need to use an appropriate encryption function before storing the data on the TOE.

OT.DataMod Protection of integrity of client application data

The TOE provides secure channels to client applications that can be used to protect the integrity of sensitive data (such as data to be signed, authentication/authorisation data or public key certificates) during transmission between the client application and the TOE.

OT.ImportExport Secure import and export of keys

The TOE allows import and export of secret keys only by using a secure method that protects the confidentiality and integrity of the data during transmission – in particular, secret keys must be exported only in encrypted form (it is not sufficient to rely on properties of a secure channel to provide the protection: the key itself must be encrypted). The TOE also allows individual secret keys under its control to be identified as non-exportable, in which case any attempt to export them will be rejected automatically. Public keys may be imported and exported in a manner that protects the integrity of the data during transmission.

Assigned keys cannot be imported or exported.

OT.Backup Secure backup of user data

Any method provided by the TOE for backing up user data, including secret keys, preserves the security of the data and is controlled by authorised Administrators. The secure backup process preserves the confidentiality and integrity of the data during creation, transmission, storage and restoration of the backup data. Backups also preserve the integrity of the attributes of keys.

OT.RNG Random number quality

Random numbers generated and provided to client applications for use as keys, authentication/authorisation data, or seed data for another random number generator that is used for these purposes shall meet a defined quality metric in order to ensure that random numbers are not predictable and have sufficient entropy.

OT.TamperDetect Tamper Detection

The TOE shall provide features to protect its security functions against tampering. In particular the TOE shall make any physical manipulation within the scope of the intended environment (adhering to OE.Env) detectable for the administrators of the TOE.

OT.FailureDetect Detection of TOE hardware or software failures

The TOE detects faults that would cause some other security property to be weakened or to fail, including:

- Environmental conditions outside normal operating range (including temperature and power)
- Failures of critical TOE hardware components (including the RNG)
- Corruption of TOE software.

On detection of a fault, the TOE takes action to maintain its security and the security of the data that it contains and controls.

OT.Audit Generation of audit trail

The TOE creates audit records for security-relevant events, recording the event details and the subject associated with the event. The TOE ensures that the audit records are protected against accidental or malicious deletion or modification of records by providing tamper protection (~~either prevention or~~ detection) for the audit log.

4.2 Security Objectives for the Operational Environment

The following security objectives relate to the TOE environment. This includes client applications as well as the procedure for the secure operation of the TOE.

OE.ExternalData Protection of data outside TOE control

Where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities shall provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment. This includes protection of data that is exported from, or imported to, the TOE (such as audit data and encrypted keys).

In particular, any backups of the TOE and its data shall be maintained in a way that ensures appropriate controls over making backups, storing backup data, and using backup data to restore an operational TOE. The number of sets of backup data shall not exceed the minimum needed to ensure continuity of the TSP service. The ability to restore a TOE to an operational state from backup data shall require at least dual person control (i.e. the participation and approval of more than one authenticated administrator).

OE.Env Protected operating environment

The TOE shall operate in a protected environment that limits physical access to the TOE to authorised Administrators. The TOE software and hardware environment (including client applications) shall be installed and maintained by Administrators in a secure state that mitigates against the specific risks applicable to the deployment environment, including (where applicable):

- Protection against loss or theft of the TOE or any of its externally stored assets

- Inspections to deter and detect tampering (including attempts to access side-channels, or to access connections between physically separate parts of the TOE, or parts of the hardware appliance)
- Protection against the possibility of attacks based on emanations from the TOE (e.g. electromagnetic emanations) according to risks assessed for the operating environment
- Protection against unauthorised software and configuration changes on the TOE and the hardware appliance
- Protection to an equivalent level of all instances of the TOE holding the same assets (e.g. where a key is present as a backup in more than one instance of the TOE).

OE.DataContext Appropriate use of TOE functions

Any client application using the cryptographic functions of the TOE shall ensure that the correct data are supplied in a secure manner (including any relevant requirements for authenticity, integrity and confidentiality). For example, when creating a digital signature over a DTBS the client application shall ensure that the correct (authentic, unmodified) DTBS/R is supplied to the TOE, and shall correctly and securely manage the signature received from the TOE; and when certifying a public key the client application shall ensure that necessary checks are made to prove possession of the corresponding private key. The client application may make use of appropriate secure channels provided by the TOE to support these security requirements. Where required by the risks in the operational environment a suitable entity (possibly the client application) shall perform a check of the signature returned from the TOE, to confirm that it relates to the correct DTBS.

Client applications shall be responsible for any required logging of the uses made of the TOE services, such as signing (or sealing) events.

Similar requirements shall apply in local use cases where no client application need be involved, but in which the TOE and its user data (such as keys used for signatures) need to be configured in ways that will support the need for security requirements such as sole control of signing keys.

Appropriate procedures shall be defined for the initial creation of data and continuing operation of the TOE according to the specific risks applicable to the deployment environment and the ways in which the TOE is used.

OE.Uauth Authentication of application users

Any client application using the cryptographic services of the TOE shall correctly and securely gather identification and authentication/authorisation data from its users and securely transfer it to the TOE (protecting the confidentiality of the authentication/authorisation data as required) when required to authorise the use of TOE assets and services.

OE.AuditSupport Audit data review

The audit trail generated by the TOE will be collected, maintained and reviewed by a System Auditor according to a defined audit procedure for the TSP.

Application Note: As noted for P.Audit, the TOE is assumed to exist as part of a larger system and the System Auditor is a role within this larger system.

OE.AppSupport Application security support

Procedures to ensure the ongoing security of client applications and their data shall be defined and followed in the environment, and reflected in use of the appropriate TOE cryptographic functions and parameters, and appropriate management and administration actions on the TOE. This includes, for example, any relevant policies on algorithms, key generation methods, key lengths, key access, key import/export, key usage limitations, key activation, cryptoperiods and key renewal, and key/certificate revocation.

5 Extended Components Definition

5.1 Extended Security Functional Requirements

Refer to the extended components defined in the Extended Components Definition section of [CEN EN 419221-5], which are included in this ST with no modifications.

6 Security Requirements

6.1 Typographical Conventions

The following conventions are used in the definitions of the SFRs:

- Selections and assignments made in this ST are ***italicised and bold.***
- Iterations in this ST that are taken from the Protection Profile [CEN EN 419221-5] and indicated with a '/' after the CC component.
- Text from the Protection Profile [CEN EN 419221-5] which is not applicable to this ST is~~crossed out~~.

6.2 Security Functional Requirements

6.2.1 Cryptographic Support (FCS)

6.2.1.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to:	No other components
Dependencies:	FCS_COP.1 Cryptographic operation FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified~~cryptographic key generation algorithm~~ ***list of key generation algorithm specified in Key Generation Table below*** and specified cryptographic key sizes ***specified in Key Generation Table below*** that meet the following: ***list of standards specified in Key Generation Table below.***

Key generation algorithm	Key size(s)	Standard
Asymmetric key generation	See FCS_COP.1	FIPS 186-4 SP 800-56A
Symmetric key generation	See FCS_COP.1	Direct generation using FCS_RNG.1/DRBG

Table 3 Key Generation Table

6.2.1.2 FCS_CKM.4 Cryptographic key destruction

Hierarchical to:	No other components

Dependencies:	FCS_CKM.1 Cryptographic key generation
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *zeroisation* that meets the following: ***FIPS 140-3 Level 3***.

6.2.1.3 FCS_COP.1 Cryptographic operation

Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic key generation FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1	The TSF shall perform ***list of cryptographic operations specified in CC-endorsed Cryptographic Algorithms Table below*** in accordance with a specified cryptographic algorithm ***specified in CC-endorsed Cryptographic Algorithms Table below*** and cryptographic key sizes ***specified in CC-endorsed Cryptographic Algorithms Table below*** that meet the following: ***list of standards specified in CC-endorsed Cryptographic Algorithms Table below***.

Algorithm and mode	**Standard**	**Key sizes**	**Use**
Advanced Encryption Standard (AES) • ECB • CBC • GCM	FIPS 197 SP800-38A SP800-38D	128 bits 192 bits 256 bits	Data encryption/decryption
Advanced Encryption Standard (AES) • Key Wrapping (KW) • Key Wrapping with Padding (KWP) • GCM	SP800-38F SP800-38D	128 bits 192 bits 256 bits	Key wrapping/unwrapping (KTS)
Advanced Encryption Standard (AES) for SSH crypto • CTR	FIPS 197 SP800-38A	128 bits	Data encryption/decryption

Algorithm and mode	Standard	Key sizes	Use
• GCM	SP800-38D		
RSA OAEP	SP 800-56Brev2	2048 bits 3072 bits 4096 bits	Key transport (KTS)
CKG	SP 800-133	128 bits 192 bits 256 bits	Symmetric key generation
RSA • RSASSA-PKCS-v1_5 • RSASSA-PSS	FIPS 186-4	1024 bits (verification only) 2048 bits 3072 bits 4096 bits	Key generation Signature generation and verification
Elliptic Curve Digital Signature Algorithm (ECDSA)	FIPS 186-4	• NIST P-224, P-256, P-384, P-521 • NIST K-233, K-283, K-409, K-571 • NIST B-233, B-283, B-409, B-571 • 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
Digital Signature Algorithm (DSA)	FIPS 186-4	L = 1024 bits, N = 160 bits (verification only) L = 2048 bits, N = 224 bits L = 2048 bits, N = 256 bits L = 3072 bits, N = 256 bits	Key generation Signature generation and verification

Algorithm and mode	Standard	Key sizes	Use
HMAC-SHA1 HMAC-SHA2	FIPS 198-1	>= 112 bits	MAC generation and verification
Advanced Encryption Standard (AES) • CMAC	SP800-38B	128 bits 192 bits 256 bits	MAC generation and verification
Diffie-Hellman (DH)	SP 800-56Arev3	MODP-2048 MODP-3072 MODP-4096 MODP-6144 MODP-8192 FB FC	Key establishment (KAS)
Elliptic Curve Diffie-Hellman (ECDH)	SP 800-56Arev3	• NIST P-224, P-256, P-384, P-521 • NIST K-233, K-283, K-409, K-571 • NIST B-233, B-283, B-409, B-571 • 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 establishment (KAS) and (KAS-SSC)
Elliptic Curve Menezes–Qu–Vanstone (ECMQV)	SP 800-56Arev3	• NIST P-224, P-256, P-384, P-521 • NIST K-233, K-283, K-409, K-571 • NIST B-233, B-283, B-409, B-571	Key establishment (KAS)

Algorithm and mode	Standard	Key sizes	Use
		• 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 Based KDF (KBKDF): • counter mode	SP 800-108	n/a	Key derivation
Secure Shell (SSH) KDF	SP 800-135rev1	n/a	Key derivation
SHA-1	FIPS 180-4	n/a	Message digest
SHA-224, SHA-256, SHA-384, SHA-512	FIPS 180-4	n/a	Message digest
SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 202	n/a	Message digest
Hash-based DRBG	SP 800-90A	344 bits	Random bit generation

Table 4 CC-endorsed Cryptographic Algorithms

6.2.1.4 FCS_RNG.1/PTRNG Generation of random numbers (AIS 31 class PTG.2)

Hierarchical to:	No other components
Dependencies:	No dependencies.
FCS_RNG.1.1/PTRNG	The TSF shall provide a ***physical*** random number generator that implements: ***(PTG.2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.*** ***(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source.***

	(PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected. ***(PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.*** ***(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered continuously. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.***
FCS_RNG.1.2/PTRNG	The TSF shall provide ***octets of bits*** that meet: ***(PTG.2.6) Test procedure A and none does not distinguish the internal random numbers from output sequences of an ideal RNG.*** ***(PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997***

Note: The Physical True RNG is only used to seed the Deterministic RNG defined in FCS_RNG.1/DRBG.

6.2.1.5 FCS_RNG.1/DRBG Generation of random numbers (AIS 20 class DRG.4)

Hierarchical to:	No other components
Dependencies:	No dependencies.
FCS_RNG.1.1/DRBG	The TSF shall provide a ***hybrid deterministic*** random number generator that implements: ***(DRG.4.1) The internal state of the RNG shall have at least 128 bits of entropy.*** ***(DRG.4.2) The RNG provides forward secrecy.*** ***(DRG.4.3) The RNG provides backward secrecy even if the current internal state is known.*** ***(DRG.4.4) The RNG provides enhanced forward secrecy on condition reseeding (every 80,000 requests).*** ***(DRG.4.5) The internal state of the RNG is seeded by an PTRNG of class PTG.2.***
FCS_RNG.1.2/DRBG	The TSF shall provide ***octets of bits*** that meet: ***(DRG.4.6) The RNG generates output for which 2^{47} strings of bit length 128 are mutually different with probability 1 - 2^{-34}*** ***(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A and NIST SP 800-22 test suite.***

Note: The second assignment in DRG.4.6 is derived from the maximum requests per reseed defined in [SP 800-90A] for an output block size of 256 bits. The

probability is derived from the formula in [AIS31, para. 337]. This corresponds to AVA_VAN.5 as specified in [AIS31, Table 13] with $k > 2^{\wedge 34}$ and $\varepsilon < 2^{\wedge -16}$.

6.2.2 Identification and Authentication (FIA)

6.2.2.1 FIA_UID.1 Timing of identification

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1	The TSF shall allow (1) Self test according to FPT_TST_EXT.1 (2) ***none*** on behalf of the user to be performed before the user is identified.
FIA_UID.1.2	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.2.2.2 FIA_UAU.1 Timing of authentication

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FIA_UAU.1.1	The TSF shall allow (1) Self-test according to FPT_TST_EXT.1, (2) Identification of the user by means of TSF required by FIA_UID.1 ***(3) none*** on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.2.2.3 FIA_UAU.6/KeyAuth_Token Re-authenticating

Hierarchical to:	No other components
Dependencies:	No dependencies.
FIA_UAU.6.1/KeyAuth_Token	The TSF shall authorise and re-authorise the user for access to a secret key under the conditions

	(1) Authorisation in order to be granted initial access to the key; and (2) ***Re-authorisation of token protected secret keys under the following conditions:*** • ***after expiry of the time period (as specified in the secret key's attributes) for which the secret key was last authorised;*** • ***after the number of uses of the secret key (as specified in the secret key's attributes) for which the secret key was last authorised has already been made;*** • ***after explicit rescinding of previous authorisation for access to the secret key.***

6.2.2.4 FIA_UAU.6/KeyAuth_CertifierKey Re-authenticating

Hierarchical to:	No other components
Dependencies:	No dependencies.
FIA_UAU.6.1/KeyAuth_CertifierKey	The TSF shall authorise and re-authorise the user for access to a secret key under the conditions (1) Authorisation in order to be granted initial access to the key; and (2) ***Re-authorisation of secret key protected by a certifier key under the following conditions:*** • ***after explicit rescinding of previous authorisation for access to the secret key.***

6.2.2.5 FIA_AFL.1 Authentication failure handling

Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication.
FIA_AFL.1.1	The TSF shall detect when ***one*** unsuccessful authentication or authorisation attempts occur related to consecutive failed authentication or authorisation attempts
FIA_AFL.1.2	When the defined number of unsuccessful authentication or authorisation attempts has been ***met***, the TSF shall block access to ***command processing*** until a time period ***of 4s*** has elapsed.

Note: this SFR applies to logical token authentication.

6.2.3 User data protection (FDP)

6.2.3.1 FDP_IFC.1/KeyBasics Subset information flow control

Hierarchical to:	No other components
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1/KeyBasics	The TSF shall enforce the Key Basics SFP on (1) subjects: all; (2) information: keys; (3) operations: all.

6.2.3.2 FDP_IFF.1/KeyBasics Simple security attributes

Hierarchical to:	No other components
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation
FDP_IFF.1.1/KeyBasics	The TSF shall enforce the Key Basics SFP based on the following types of subject and information security attributes: (1) whether a key is a secret or a public key (2) whether a secret key is an Assigned Key (3) whether channels selected to export keys are secure (4) the value of the Export Flag of a key.
FDP_IFF.1.2/KeyBasics	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: (1) Export of secret keys shall only be allowed provided that the secret key is not an Assigned Key, that the secret key is encrypted, and that a secure channel (providing authentication and integrity protection) is used for the export; (2) Public keys shall always be exported with integrity protection of their key value and attributes; (3) Keys shall only be imported over a secure channel (providing authentication and integrity protection); (4) A secret key can only be imported if it is a non-Assigned key; (5) Secret keys shall only be imported in encrypted form~~or using split-knowledge procedures requiring at least two key components to reconstruct the key, with key components supplied by at least two separately authenticated users~~.
FDP_IFF.1.3/KeyBasics	The TSF shall enforce the following additional information flow control rules: none.

FDP_IFF.1.4/KeyBasics	The TSF shall explicitly authorise an information flow based on the following rules: none.
FDP_IFF.1.5/KeyBasics	The TSF shall explicitly deny an information flow based on the following rules: (1) No subject shall be allowed to access the plaintext value of any secret key directly; (2) No subject shall be allowed to export a secret key in plaintext; (3) No subject shall be allowed to export an Assigned Key; (4) No subject shall be allowed to export a secret key without submitting the correct authorisation data for the key; (5) No subject shall be allowed to access intermediate values in any operation that uses a secret key; (6) A key with an Export Flag value marking it as non-exportable shall not be exported.

Note: The requirement "(6) Unblocking access to a key shall not allow any subject other than those authorised to access the key at the time when it was blocked." in FDP_IFF.1.2/KeyBasics in the PP is not applicable (refer FMT_MTD.1/Unblock) and, for readability purposes, has been removed in the ST.

6.2.3.3 FDP_ACC.1/KeyUsage Subset access control

Hierarchical to:	No other components
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/KeyUsage	The TSF shall enforce the Key Usage SFP on (1) subjects: all; (2) objects: keys; (3) operations: all.

6.2.3.4 FDP_ACF.1/KeyUsage Security attribute based access control

Hierarchical to:	No other components
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/KeyUsage	The TSF shall enforce the Key Usage SFP to objects based on the following: (1) whether the subject is currently authorised to use the secret key; (2) whether the subject is currently authorised to change the attributes of the secret key; (3) the cryptographic function that is attempting to use the secret key.

FDP_ACF.1.2/KeyUsage	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: (1) Attributes of a key shall only be changed by an authorised subject, and only as permitted in the Key Attributes Modification Table (2) Only subjects with current authorisation for a specific secret key shall be allowed to carry out operations using the plaintext value of that key (3) Only cryptographic functions permitted by the secret key's Key Usage attribute shall be carried out using the secret key.
FDP_ACF.1.3/KeyUsage	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none.
FDP_ACF.1.4/KeyUsage	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none.

6.2.3.5 FDP_ACC.1/Backup Subset access control

Hierarchical to:	No other components
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1/Backup	The TSF shall enforce the Backup SFP on (1) subjects: all; (2) objects: keys; (3) operations: backup, restore.

6.2.3.6 FDP_ACF.1/Backup Security attribute based access control

Hierarchical to:	No other components
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/Backup	The TSF shall enforce the Backup SFP to objects based on the following: (1) whether the subject is an administrator
FDP_ACF.1.2/Backup	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: (1) Only authorised administrators shall be able to perform any backup operation provided by the TSF to create backups of the TSF state or to restore the TSF state from a backup (2) Any restore of the TSF shall only be possible under at least dual person control, with each person being an administrator (3) Any backup and restore shall preserve the confidentiality and integrity of the secret keys, and the integrity of public keys

	(4) Any backup and restore operations shall preserve the integrity of the key attributes, and the binding of each set of attributes to its key.
FDP_ACF.1.3/Backup	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: none.
FDP_ACF.1.4/Backup	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none.

6.2.3.7 FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring.
Dependencies:	No dependencies.
FDP_SDI.2.1	The TSF shall monitor user data stored in containers controlled by the TSF for integrity errors on all keys (including security attributes), based on the following attributes: integrity protection data.
FDP_SDI.2.2	Upon detection of a data integrity error, the TSF shall (1) prohibit the use of the altered data; (2) notify the error to the user.

6.2.3.8 FDP_RIP.1 Subset residual information protection

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_RIP.1.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the deallocation of the resource from the following objects: • authorisation data; • secret keys.

6.2.4 Trusted path/channels (FTP)

6.2.4.1 FTP_TRP.1/Local Trusted Path

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_TRP.1.1/Local	The TSF shall provide a communication path between itself and local client applications that is logically distinct from other communication paths and

	provides assured authentication of its end points and protection of the communicated data from modification and disclosure.
FTP_TRP.1.2/Local	The TSF shall permit ***Local and Local Embedded CodeSafe client applications*** to initiate communication via the trusted path.
FTP_TRP.1.3/Local	The TSF shall require the use of the trusted path for ***accessing services offered by the TSF.***

Application Note

Application Note 29 from Protection Profile [CEN EN 419221-5] specifies: *"FTP_TRP.1/Local must be completed in the Security Target to identify the local client applications and to reflect the way that the TOE communicates with them, and to justify the security of this communication path. Where the TOE and local client applications are located within the physical boundary of the same hardware appliance (e.g. local applications running on a server and communicating with a PCI card on the server's internal PCI bus), then the trusted path may be mapped in the Security Target to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment)."*

Due to the local nature of the communication between the TOE and the local application (within the hardware appliance) or the local embedded CodeSafe application, the physical configuration ensures authentication of the end points, as well as integrity and confidentiality of data. No cryptographically secure trusted channel is required. Note that before being allowed to run on the TOE, the signature of Local Embedded Codesafe Applications is verified by the TOE.

Despite not being strictly required by the SFR, the TOE implements means to ensure a dedicated communication path between the container (where the CodeSafe application resides) and the nCoreAPI service of the TOE, by means of an IPC (Unix domain socket) link. The IPC link is uniquely associated to the container identity and, once established, the CodeSafe application can issue nCoreAPI commands. These nCoreAPI commands are only executed after successful validation by the TOE using *SEECert* certificates.

Note that secure channels may also be established between external and local (embedded) client applications (for example via a dedicated SSH channel), but these channels are not covered within the scope of this Security Target.

6.2.5 Protection of the TSF (FPT)

6.2.5.1 FPT_STM.1 Reliable time stamps

Hierarchical to:	No other components
Dependencies:	No dependencies.
FPT_STM.1.1	The TSF shall be able to provide reliable time stamps.

6.2.5.2 FPT_TST_EXT.1 Basic TSF Self Testing

Hierarchical to:	No other components
Dependencies:	No dependencies.

FPT_TST_EXT.1.1	The TSF shall run a suite of the following self-tests during initial start-up (or power-on) and ***periodically during normal operation*** to demonstrate the correct operation of the TSF:
	• At initial start-up (or power-on):
	  ○ Software/firmware integrity test
	  ○ Cryptographic algorithm tests
	  ○ Random number generator tests
	• ***Periodically during normal operation:***
	  ○ ***SP 800-90B health tests***
	  ○ ***AIS 31 online test procedure***

6.2.5.3 FPT_PHP.1 Passive detection of physical attack

Hierarchical to:	No other components
Dependencies:	No dependencies.
FPT_PHP.1.1	The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.
FPT_PHP.1.2	The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

Application Note

Application Note 33 from [CEN EN 419221-5] states that "passive detection of a physical attack is typically achieved by using physical seals and an appropriate physical design of the TOE that allows the TOE administrator to verify the physical integrity of the TOE as part of a routine inspection procedure. Because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 for this TOE is equivalent to the physical security mechanisms for tamper detection and response required by section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements for each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3. (Cf. refinement of AVA_VAN.5 in section 7.4.1.)"

It shall be noted that in this Security Target, physical security requirements are mapped to FIPS 140-3, which aligns with ISO/IEC 19790:2012(E).

6.2.5.4 FPT_PHP.3 Resistance to physical attack

Hierarchical to:	No other components
Dependencies:	No dependencies.
FPT_PHP.3.1	The TSF shall resist ***physical penetration attempts*** to the ***hard opaque potted enclosure*** by responding automatically such that the SFRs are always enforced.

Application Note:

Application Note 34 from [CEN EN 419221-5] states that "This SFR is linked to the requirements for passive detection of physical attacks in FPT_PHP.1, and should identify the relevant responses of the TOE involved in meeting the key zeroisation requirements of ISO/IEC 19790:2012 Security Level 3. As in the case of FPT_PHP.1, because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.3 for this TOE is equivalent to the level of assessment for this aspect of tamper detection and response required for section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements by each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3. (Cf. refinement of AVA_VAN.5 in section 7.4.1.)".

It shall be noted that in this Security Target, physical security requirements are mapped to FIPS 140-3, which aligns with ISO/IEC 19790:2012(E).

6.2.5.5 FPT_FLS.1 Failure with preservation of secure state

Hierarchical to:	No other components
Dependencies:	No dependencies.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: (1) Self-test according to FPT_TST_EXT.1 fails; (2) Environmental conditions are outside normal operating range (including temperature and power); (3) Failures of critical TOE hardware components (including the RNG) occurs; (4) Corruption of TOE software occurs; (5) ***none***

6.2.6 Security Management (FMT)

6.2.6.1 FMT_SMR.1 Security roles

Hierarchical to:	No other components.

Dependencies:	FIA_UID.1 Timing of identification.
FMT_SMR.1.1	The TSF shall maintain the roles Administrator **(i.e. *Platform Crypto Officer (PCO), nShield Security Officer (NSO))*, *Local Client Application***, Key User, ***none.***
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

Application Note:

- Platform Crypto Officer (PCO) is responsible for administration tasks of the HSM platform. The PCO has access to the local host machine within the physical protected environment. Authenticated access to administration TOE services is provided by dedicated SSH channels.
- nShield Security Officer (NSO) is responsible for overall management of the Security World.

6.2.6.2 FMT_SMF.1 Security management functions

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: (1) Modifying attributes of keys; (2) Export and deletion of the audit data, which can take place only under the control of the Administrator role; (3) ***backup and restore functions;*** (4) ***key import function;*** (5) ***key export function.*** ***(6) Get platform information (Setup Service)*** ***(7) Factory reset (Setup factorystate)*** ***(8) ssh secure channel administration (SSHAdmin service)*** ***(9) Software update (Updater Service)*** ***(10) CodeSafe Application loading and signature verification (Launcher Service)***

Application Note:

The requirement "(1) Unblock of access due to authentication or authorisation failures" in the PP is not applicable (refer FMT_MTD.1/Unblock) and, for readability purposes, has been removed in the ST.

6.2.6.3 FMT_MTD.1/Unblock Management of TSF data

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/Unblock	The TSF shall restrict the ability to unblock the ***none*** to ***none.***

Application note: The TOE does not block as a result of failed authentications, but imposes a 4 seconds delay. See FIA_AFL.1.

6.2.6.4 FMT_MTD.1/AuditLog Management of TSF data

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1/AuditLog	The TSF shall restrict the ability to control export and deletion of the audit log records to the Administrator role.

6.2.6.5 FMT_MSA.1/GenKeys Management of security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1/GenKeys	The TSF shall enforce the Key Usage SFP to restrict the ability to modify the security attributes ***specified in the Key Attributes Modification Table*** to ***subjects, objects, and operations among subjects and General Keys as specified in the Key Attributes Modification Table***.

6.2.6.6 FMT_MSA.1/AKeys Management of security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1/AKeys	The TSF shall enforce the Key Usage SFP to restrict the ability to modify the security attributes ***specified in the Key Attributes Modification Table*** to ***subjects, objects, and operations among subjects and Assigned Keys as specified in the Key Attributes Modification Table***.

Key Attribute (MSA.1)	**Assigned Key**	**General Key**
Key ID	Cannot be modified	Cannot be modified

Key Attribute (MSA.1)	Assigned Key	General Key
Key type	Cannot be modified	Cannot be modified
Authorisation data	For Token protected keys: Modified only when modification operation includes successful validation of current (pre-modification) authorisation data. For Certifier protected keys: cannot be modified.	Modified only when modification operation includes successful validation of current (pre-modification) authorisation data, or by an Administrator
Re-authorisation conditions	Cannot be modified	-
Key usage	Cannot be modified	-
Export flag	Cannot be modified	-
Assigned flag	Cannot be modified	~~Can be modified only by Administrator, and only to change from non-assigned to assigned~~ Cannot be modified
Integrity protection data	Cannot be modified by users (maintained automatically by TSF)	Cannot be modified by users (maintained automatically by TSF)

Table 5 Key Attributes Modification Table

6.2.6.7 FMT_MSA.3/Keys Static attribute initialisation

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1/Keys	The TSF shall enforce the Key Usage SFP to provide ***restrictive*** default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/Keys	The TSF shall allow the ***authorised identified roles, according to the constraints in the Key Attributes Initialisation Table*** to specify alternative initial values to override the default values when an object or information is created.

Key Attribute (MSA.1)	Assigned Key	General Key
Key ID	Initialised by generation process	Initialised by generation process
Key type	Initialised by generation process	Initialised by generation process
Authorisation data	Initialised by creator during generation	Initialised by creator during generation
Re-authorisation conditions	Initialised by Administrator during generation	-
Key usage	Initialised by creator during generation	-
Export flag	False (i.e. no export allowed)	-
Assigned flag	Initialised by generation process	Non-assigned
Integrity protection data	Initialised automatically by TSF	Initialised automatically by TSF

Table 6 Key Attributes Initialisation Table

6.2.7 Security Audit Data Generation (FAU)

6.2.7.1 FAU_GEN.1 Audit data generation

Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable time stamps
FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the not specified level of audit; and c) Startup of the TOE; d) Shutdown of the TOE; e) Cryptographic key generation (FCS_CKM.1); f) Cryptographic key destruction (FCS_CKM.4); g) Failure of the random number generator (FCS_RNG.1); h) Authentication and authorisation failure handling (FIA_AFL.1): all unsuccessful authentication or authorisation attempts,~~the reaching of the threshold for the unsuccessful authentication or authorisation attempts and the blocking actions taken~~; i) All attempts to import or export keys (FDP_IFF.1/KeyBasics); j) All modifications to attributes of keys (FDP_ACF.1/KeyUsage, FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys);

	k) Backup and restore (FDP_ACF.1/Backup): use of any backup function, use of any restore function, unsuccessful restore because of detection of modification of the backup data; l) Integrity errors detected for keys (FDP_SDI.2); m) Failures to establish secure channels (FTP_TRP.1/Local); n) Self-test completion (FPT_TST_EXT.1); o) Failures detected by the TOE (FPT_FLS.1); p) All administrative actions (FMT_SMF.1, FMT_MSA.1 (all iterations), FMT_MSA.3/Keys); ~~q) Unblocking of access (FMT_MTD.1/Unblock);~~ r) Modifications to audit parameters (affecting the content of the audit log) (FAU_GEN.1); ***s) None***
FAU_GEN.1.2	The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST: ***none***.

6.2.7.2 FAU_GEN.2 User identity association

Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification
FAU_GEN.2.1	For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.2.7.3 FAU_STG.2 Guarantees of audit data availability

Hierarchical to:	FAU_STG.1 Protected audit trail storage
Dependencies:	FAU_GEN.1 Audit data generation
FAU_STG.2.1	The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
FAU_STG.2.2	The TSF shall be able to ***detect*** unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.2.3	The TSF shall ensure that all stored audit records will be maintained when the following conditions occur: ***audit storage exhaustion.***

6.3 Security Assurance Requirements

The security assurance requirement level is EAL4 augmented with AVA_VAN.5 as required by [CEN EN 419221-5, section 6.4].

The refinements made to the SARs can be found in [CEN EN 419221-5, section 6.4.1].

Additionally, this Security Target augments the assurance requirements with ALC_FLR.2.

7 TOE Summary Specification

This section provides a summary of how the TOE meets the SFRs.

For readability, closely related SFRs are grouped in Security Functions provided by the TOE, and a description of the Security Functions is provided.

7.1 Key management

Key attributes

The TOE maintains an Access Control List (ACL) associated with a key. ACLs are ordered lists of permission groups, which consists of one or more actions, and optionally restrictions.

The ACL of a private / secret key is specified at key generation and the TOE will enforce restrictive default values, meaning that if a certain value or restriction is not specified, the TOE will assume the restrictive value.

When secret / private keys are generated according to the TOE guidance documentation, the requirements in the Key Usage SFP and the Key Basics SFP are enforced. In this way, the concept of Assigned Key is enforced by the TOE with a very specific ACL configuration, instead of a dedicated *Assigned Key* flag. The Assigned Key ACL configuration enforces that:

- The ACL is non-modifiable,
- The key cannot be exported or used in a key derivation operation,
- The key is non-recoverable.

Key generation

The TOE provides key generation functions using the on board FIPS 140-3 Level 3 and AIS 31 compliant Random Number Generator.

When a new key pair is generated, the TOE also generates a signed key generation certificate, which contains information about the ACL, hash of the key, and key generation parameters. This way, public keys are exported with integrity protection of their key value and attributes, as required by FDP_IFF.1.2/KeyBasics (2).

Key storage and backup

Keys are stored in a secure container, a Key Blob, which is under the control of the TOE and ensures the confidentiality and integrity of the key and ACL.

The TOE supports the following backup operations:

- *"backup create"* by creating a new Security World,
- *"backup restore"* by loading an already existing Security World into an HSM.

Key import and export

Keys can be exported only if allowed by the key ACL.

The ACL configuration for Assigned Keys explicitly disables key export. The ACL configuration for General Keys allows export only in encrypted form.

Only General keys can be imported. General keys can only be imported in encrypted form.

Note: key export means that the key is no longer under the exclusive control of the TOE, as opposed to external storage and backup, in which the keys remain under exclusive control of the TOE.

Key destruction

Internal keys are deallocated by overwriting the object in volatile memory with a zero-bit pattern.

Module keys in persistent storage are zeroized with zero-bit pattern or random data.

7.2 Authorisation

The TOE protects keys and sensitive functions from unauthorised access. For the defined user roles, the following holds:

- The Client Application role (mapped to subject S.Application) can be of two types:
-
    - local applications (physically residing within the same hardware appliance and communicating with the TOE via the PCIe interface), and/or
    - local embedded applications (i.e. CodeSafe applications, residing within the TOE hardware boundary).

  As described in [CEN EN 419221-5, FTP_TRP.1/Local] and [CEN EN 419221-5, Application Note 29], the authentication of local (embedded) clients applications is ensured by the local nature of the communication between the TOE and the application. Note that before being allowed to run on the TOE, the signature of Local Embedded Codesafe Applications is verified by the TOE.
- The Platform Crypto Officer (PCO) role (mapped to subject S.Admin) is authenticated by means of dedicated SSH secure channel private keys to access platform administration services (SSHAdmin, Updater, Setup, Monitor and Launcher services). Note that the PCO has physical access to the physical protected environment where the TOE is installed, thus gaining local access to the local host machine.
- The nShield Security Officer (NSO) role (mapped to subject S.Admin) is undertaken by holders of the Administrator Card Set (ACS). Access to Security World administrative functions requires a quorum of Administrator Card Set (ACS) holders with passphrases presented to the TOE. In addition,

note that Security World administrative functions are provided by the nCoreAPI service, which can be accessed by the NSO only after successful establishment of an SSH secure channel, authenticated by means of a dedicated private key.

- The Key User (KU) role (mapped to subject S.User) is represented by owners of application keys, e.g. Signing keys. Access to usage of a key, e.g. for signing, encryption or any other cryptographic functionality provided by the nCoreAPI service, requires one of the following:
-
    - For token protected secret keys, a quorum of Operator Card Set (OCS) or a Softcard with passphrases shall be presented to the TOE.
    - For certifier protected secret keys, the Certifier Key specified in the Access Control List (ACL) of the secret key must be loaded in the TOE.

  In case of failed attempt to load a token from a smartcard or softcard with a passphrase, the TOE enforces a 4 seconds delay to prevent brute force attacks.

  Token protected keys support time and use limits imposed in the ACL of the key. Authorisation to use a key can be removed by deallocating the key handle through the Cmd_Destroy call.

  In addition, note that the Key User role gains access to functions provided by the nCoreAPI service of the TOE, via the Client Application authenticated as already described above.

7.3 Cryptographic functions

The TOE provides a wide range of cryptographic functions available to applications through the nCoreAPI. Refer to FCS_COP.1 for a detailed list of algorithms.

7.4 Random number generation

The random number generator used by the TOE is composed of two main blocks:

- A deterministic RNG providing 256 bits of security, compliant with NIST SP 800-90A Hash-based DRBG and DRG.4
- A physical true RNG compliant with PTG.2 and SP 800-90B, which provides entropy input to seed the deterministic RNG.

Only random numbers output from the deterministic RNG are used by the TOE for a variety of functions, mainly key generation, random values such as IVs, nonces and random number generation service provided by the nCoreAPI to requesting applications.

7.5 Audit

The TOE is configured to generate audit logs of all the relevant events specified in FAU_GEN.1.1. The on board Real Time Clock (RTC) is used to provide the log timestamp. This RTC is also available through the API as a service to other timestamping applications.

Once the audit function is enabled by the Administrator during TOE initial configuration, the log entries generated by the nCoreAPI Service are asynchronously and automatically sent out of the TOE (to the Hardserver), without any further human intervention. It is the responsibility of the Hardserver (out of scope) to send the audit logs to a Syslog server.

The TOE uses the same memory buffer to queue incoming nCoreAPI commands and log entries to be sent to the Hardserver. The TOE throttles inputs if required, preventing exhaustion of the internal/temporary buffers.

Audit log entries generated by other TOE services are locally stored and available to the Platform Administrator via the Monitor service.

The TOE maintains an audit signing key pair which is used to sign the audit trail Signature Blocks. The public key is available to external parties for verification of the audit trail integrity and origin.

7.6 Physical protection

The TOE is enclosed in a hard opaque potting enclosure, meeting FIPS 140-3 level 3 physical requirements, to protect itself against physical attacks.

7.7 Self tests

The TOE implements self-tests and monitoring capabilities to ensure it is operating securely.

At Power-up

- Firmware integrity verification.
- AES encryption and decryption Known Answer Test
- AES CMAC Known Answer Test
- Triple-DES encryption and decryption Known Answer Test
- Triple-DES CBC-MAC Known Answer Test
- DSA signature generation and verification Known Answer Test
- ECDSA signature generation and verification pair-wise consistency test
- RSA signature generation and verification Known Answer Test
- RSA encryption and decryption Known Answer Test
- HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 Known Answer Test

- Primitive "Z" for Diffie-Hellman Known Answer Test
- Primitive "Z" for EC Diffie-Hellman and EC MQV Known Answer Test
- KDF in CTR mode Known Answer Test
- DRBG Known Answer Test
- SHA-1 Known Answer Test

During normal operation

- Voltage and temperature monitoring, which trigger a tamper response in the case of out of range values.
- AIS 31 online and total failure tests on the physical true RNG.
- SP 800-90B health tests on the physical true RNG.

7.8 Secure channel

The local nature of the communication between the TOE and local applications (either residing on the host machine or CodeSafe applications residing within the TOE hardware boundary) ensures integrity and confidentiality.

However, despite not being strictly necessary, the TOE uses SSH secure channels for all communication with local applications residing on the host machine, providing mutual authentication, confidentiality and integrity. The channels use the following ciphersuites:

- key exchange protocol `ecdh-sha2-nistp256`
- data encryption with `aes128-gcm`

7.9 Mapping between Security Functions and SFRs

	Audit	Authorisation	Key management	Physical protection	Self tests	Secure channel	Cryptographic functions	Random number generation
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_STG.2	X							
FCS_CKM.1			X					

	Audit	Authorisation	Key management	Physical protection	Self tests	Secure channel	Cryptographic functions	Random number generation
FCS_CKM.4			X					
FCS_COP.1							X	
FCS_RNG.1/PTRNG			X					X
FCS_RNG.1/DRBG			X					X
FIA_UID.1		X						
FIA_UAU.1		X						
FIA_UAU.6/KeyAuth_Token		X						
FIA_AFL.1		X						
FIA_UAU.6/KeyAuth_CertifierKey		X						
FDP_IFC.1/KeyBasics			X					
FDP_IFF.1/KeyBasics			X					
FDP_ACC.1/KeyUsage		X	X					
FDP_ACF.1/KeyUsage		X	X					
FDP_ACC.1/Backup			X					
FDP_ACF.1/Backup			X					
FDP_SDI.2			X					
FDP_RIP.1			X					
FMT_SMR.1		X						
FMT_SMF.1	X		X					
~~FMT_MTD.1/Unblock (n/a)~~								
FMT_MTD.1/AuditLog	X							
FMT_MSA.1/GenKeys			X					
FMT_MSA.1/AKeys			X					
FMT_MSA.3/Keys			X					

	Audit	Authorisation	Key management	Physical protection	Self tests	Secure channel	Cryptographic functions	Random number generation
FPT_STM.1	X							
FPT_TST_EXT.1					X			
FPT_PHP.1				X				
FPT_PHP.3				X				
FPT_FLS.1					X			
FTP_TRP.1/Local		X				X		
~~FTP_TRP.1/External~~		~~X~~				~~X~~		

8 Rationales

8.1 Security Objectives Rationale

8.1.1 Security Objectives Coverage

The table below shows the mapping of Threats, Organisational Security Policies and Assumptions to Security Objectives for the TOE and for the TOE Environment.

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit	OE.ExternalData	OE.Env	OE.DataContext	OE.AppSupport	OE.Uauth	OE.AuditSupport
T.KeyDisclose	X		X				X		X	X		X			X	X				
T.KeyDerive		X									X									
T.KeyMod			X						X	X		X								
T.KeyMisuse				X	X															
T.KeyOveruse						X														
T.DataDisclose							X										X	X		
T.DataMod								X									X	X		

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit	OE.ExternalData	OE.Env	OE.DataContext	OE.AppSupport	OE.Uauth	OE.AuditSupport
T.Malfunction													X							
P.Algorithms		X																		
P.KeyControl	X	X		X	X	X			X	X										
P.RNG											X									
P.Audit														X						
A.ExternalData															X					
A.Env																X				
A.DataContext																	X			
A.AppSupport																		X		
A.UAuth																			X	
A.AuditSupport																				X

Table 7 Security Problem Definition mapping to Security Objectives

8.1.2 Security Objectives Sufficiency

The following paragraphs describe the rationale for the sufficiency of the Security Objectives relative to the Threats, OSPs and Assumptions.

8.1.2.1 Threats

T.KeyDisclose is addressed by the requirement in OT.PlainKeyConf to keep plaintext secret keys unavailable, and this is supported in terms of controls over key attributes (which might threaten the confidentiality of the key if modified) in OT.KeyIntegrity. The confidentiality of secret keys that are exported is protected partly by the use of a secure channel as described in OT.DataConf and the requirements for import and export in OT.ImportExport (including the requirement to export secret keys only in encrypted form, or to be able to exclude the export of a key entirely). Physical tamper protection of the keys is provided by OT.TamperDetect (supported by an appropriate inspection procedure as required in OE.Env). Protection of secret key confidentiality during backup is ensured by OT.Backup. The environment also contributes to maintaining secret key confidentiality by protecting any versions of a secret key that may exist outside the TOE, as in OE.ExternalData, and by protecting the operation of the TOE itself by providing a secure environment, as in OE.Env.

T.KeyDerive is addressed by the choice of algorithms that have been endorsed for the appropriate purposes, and this is described in OT.Algorithms. Where keys are generated by the TOE then the use of a suitable random number generator is required by OT.RNG in order to mitigate the risk that an attacker can guess or deduce the key value.

T.KeyMod is addressed by requiring integrity protection of secret and public keys, and their critical attributes in OT.KeyIntegrity, and by requiring use of secure channels that protect integrity if a key is imported or exported (OT.ImportExport). Protection of key integrity during backup is ensured by OT.Backup. Physical tamper protection of the keys is provided by OT.TamperDetect (supported by an appropriate inspection procedure as required in OE.Env).

T.KeyMisuse raises the possibility of a secret key being used for an unintended and unauthorised purpose, and is addressed by the requirement in OT.Auth for the TOE to carry out an authorisation check before using a secret key. OT.KeyUseConstraint expands on this to set out requirements for the granularity of authorisation.

T.KeyOveruse is concerned with the possibility that more uses may be made of an authorised key than were intended, and this is addressed by the requirements of OT.KeyUseScope which requires controls to be specified and enforced for any re-authorisation conditions that the TOE allows a user to define.

T.DataDisclose is concerned with the transmission of data between client applications and the TOE~~, or between separate parts of the TOE where the transmission passes through an insecure environment~~. This is addressed by OT.DataConf, which requires the TOE to provide secure channels to protect such communications. The appropriate use of such channels is a requirement for the environment as expressed in OE.DataContext, as is the use of appropriate procedures in OE.AppSupport.

T.DataMod is concerned with the possibility of unauthorised modification of data transmitted between a client application and the TOE, and this is addressed by OT.DataMod which requires that the TOE provides secure channels that can be used to protect the integrity of data that they carry. As with T.DataDisclose, the appropriate use of such channels is a requirement for the environment as expressed in OE.DataContext, as is the use of appropriate procedures in OE.AppSupport.

T.Malfunction is addressed by the requirement in OT.FailureDetect for the TOE to detect certain types of fault.

8.1.2.2 Organisation Security Policies

P.Algorithms requires the use of key generation and other cryptographic functions that are endorsed by appropriate authorities, and this is addressed by OT.Algorithms.

P.KeyControl requires that the TOE can provide controls and support a key lifecycle to ensure that secret keys can be reliably protected against use by those other than the owner of the key, and that the keys can be confined to use for certain cryptographic functions. This is addressed by a combination of TOE objectives as follows:

- OT.PlainKeyConf protects the value of the secret key to prevent the possibility of it being used by unauthorised subjects
- OT.Algorithms ensures that endorsed algorithms that employ and support suitable properties and procedures are provided by the TOE
- OT.Auth, OT.KeyUseConstraint and OT.KeyUseScope ensure that the TOE can provide well-defined limits on the use of a key when it is authorised (as described above for T.KeyMisuse and T.KeyOveruse
- OT.ImportExport and OT.Backup ensure protection of keys when they are transmitted outside the TOE to client applications or for backup purposes, including the prevention of export of Assigned Keys.

P.Audit requires the TOE to provide an audit trail and this is addressed directly by OT.Audit (which includes protection of the audit records).

8.1.2.3 Assumptions

Each of the Assumptions is directly matched by a Security Objective for the operational environment. The wording of each objective for the operational environment includes the wording of each assumption, and no further rationale is therefore given here.

8.2 Security Requirements Rationale

8.2.1 Security Requirements Coverage

The table below summarises the mapping of Security Objectives for the TOE to SFRs.

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FCS_CKM.1		X												
FCS_CKM.4	X													
FCS_COP.1		X												
FCS_RNG.1/PTRNG											X			
FCS_RNG.1/DRBG											X			
FIA_UID.1				X										
FIA_UAU.1				X										
FIA_AFL.1				X										
FIA_UAU.6/KeyAuth_Token				X		X								
FIA_UAU.6/KeyAuth_CertifierKey				X		X								

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FDP_IFC.1/KeyBasics	X				X				X					
FDP_IFF.1/KeyBasics	X		X		X				X					
FDP_ACC.1/KeyUsage					X	X								
FDP_ACF.1/KeyUsage					X	X								
FDP_ACC.1/Backup										X				
FDP_ACF.1/Backup										X				
FDP_SDI.2			X											
FDP_RIP.1	X				X									
FTP_TRP.1/Local			X	X			X	X	X					
~~**FTP_TRP.1/External**~~			~~X~~	~~X~~			~~X~~	~~X~~	~~X~~					
FPT_STM.1														X
FPT_TST_EXT.1													X	
FPT_PHP.1												X		
FPT_PHP.3												X		

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit
FPT_FLS.1													X	
FMT_SMR.1				X										X
FMT_SMF.1				X										X
~~FMT_MTD.1/Unblock~~				~~X~~										
FMT_MTD.1/AuditLog														X
FMT_MSA.1/GenKeys					X									
FMT_MSA.1/AKeys					X									
FMT_MSA.3/Keys					X									
FAU_GEN.1														X
FAU_GEN.2														X
FAU_STG.2														X

Table 8 TOE Security Objectives mapping to SFRs

8.2.2 Security Requirements Sufficiency

OT.PlainKeyConf is addressed by the requirements in the Key Basics SFP defined in FDP_IFC.1/KeyBasics and FDP_IFF.1/KeyBasics (especially FDP_IFF.1.5/KeyBasics). Secure destruction of keys according to FCS_CKM.4 protects the key value at the end of its lifetime. FDP_RIP.1 protects secret keys from being accessed after they have been deallocated.

OT.Algorithms is addressed by the need to use endorsed standards for FCS_COP.1 (SOGIS and NIST) and the use of an appropriate random number generator in FCS_CKM.1.

OT.KeyIntegrity is addressed primarily by FDP_SDI.2 which requires integrity protection of keys and their attributes by the TOE. FDP_IFF.1/KeyBasics requires that any importing or exporting of keys requires the use of secure channels and integrity protection (see FTP_TRP.1/Local).

OT.Auth is addressed by FIA_UID.1, FIA_UAU.1 and FIA_AFL.1 for Administrator. Authorisation for client applications is provided by the requirements for authentication of endpoints in FTP_TRP.1/Local. Authorisation for the use of secret keys is addressed by FIA_UAU.6/KeyAuth_Token and FIA_UAU.6/KeyAuth_Certifier.

OT.KeyUseConstraint is addressed by the requirements for well-defined (and securely initialised) key attributes in FMT_MSA.1/GenKeys, FMT_MSA.1/AKeys, and FMT_MSA.3/Keys, and the application of the attributes to operate constraints on the use of keys in FDP_IFC.1/KeyBasics, FDP_IFF.1/KeyBasics, FDP_ACC.1/KeyUsage and FDP_ACF.1/KeyUsage. FDP_RIP.1 protects authorisation data (which enables a key to be used) from being accessed after it has been deallocated.

OT.KeyUseScope is addressed by the Key Usage SFP in FDP_ACC.1/KeyUsage and FDP_ACF.1/KeyUsage and by the re-authorisation conditions for use of a secret key specified in FIA_UAU.6/KeyAuth_Token and FIA_UAU.6/KeyAuth_Certifier.

OT.DataConf is addressed by the authentication and confidentiality requirements for secure channels in FTP_TRP.1/Local.

OT.DataMod is addressed by the authentication and integrity requirements for secure channels in FTP_TRP.1/Local.

OT.ImportExport is addressed by the requirements for the use of secure import/export through a secure channel and restrictions on how keys are imported and exported to protect confidentiality and integrity in the Key Basics SFP in FDP_IFC.1/KeyBasics and FDP_IFF.1/KeyBasics, and by the requirements on the secure channels themselves in FTP_TRP.1/Local.

OT.Backup separates out the requirements for any backup and restore properties that the TOE may provide and is addressed directly by the Backup SFP in FDP_ACC.1/Backup and FDP_ACF.1/Backup.

OT.RNG is addressed by the requirement in FCS_RNG.1/PTRNG and FCS_RNG.1/DRBG for a true physical random number generator and deterministic random bit generator which meet AIS 31 PTG.2 and DRG.4 classes.

OT.TamperDetect is addressed by the requirement for passive tamper detection in FPT_PHP.1 and the tamper response mechanisms in FPT_PHP.3.

OT.FailureDetect is addressed by the self-test requirements of FPT_TST_EXT.1 and secure failure requirements of FPT_FLS.1.

OT.Audit is addressed in terms of basic creation of audit records by the requirements for audit record generation in FAU_GEN.1 and FAU_GEN.2 and provision of timestamps for use in audit records in FPT_STM.1. Protection of the audit trail is ensured by FAU_STG.2, FMT_MTD.1/AuditLog and FMT_SMF.1. Support for the Administrator role that controls export and deletion of audit records from the TOE is required by FMT_SMR.1.

8.2.3 SFR Dependencies

The dependencies between SFRs are addressed as shown in the following Table. Where a dependency is not met in the manner defined in [CC2] then a rationale is provided for why the dependency is unnecessary or else met in some other way.

Requirement	Dependencies	Fulfilled by
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1 FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1 See also note below on key attributes during import or export.
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FCS_CKM.1 FCS_CKM.4 See also note below on key attributes during import or export.
FCS_RNG.1/PTRNG	No dependencies	--

Requirement	Dependencies	Fulfilled by
FCS_RNG.1/DRBG		
FIA_UID.1	No dependencies	--
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_UAU.6/KeyAuth_Token FIA_UAU.6/KeyAuth_Certifier	No dependencies	--
FDP_IFC.1/KeyBasics	FDP_IFF.1	FDP_IFF.1/KeyBasics
FDP_IFF.1/KeyBasics	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1/KeyBasics FMT_MSA.3/Keys
FDP_ACC.1/KeyUsage	FDP_ACF.1	FDP_ACF.1/KeyUsage
FDP_ACF.1/KeyUsage	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/KeyUsage FMT_MSA.3/Keys
FDP_ACC.1/Backup	FDP_ACF.1	FDP_ACF.1/Backup
FDP_ACF.1/Backup	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Backup The dependency on FMT_MSA.3 is not relevant in this case since the attribute used in FDP_ACF.1/Backup is determined by the ability of the user to authenticate as an administrator according to FIA_UAU.1.
FDP_SDI.2	No dependencies	--
FDP_RIP.1	No dependencies	--

Requirement	Dependencies	Fulfilled by
FTP_TRP.1/Local	No dependencies	--
~~FTP_TRP.1/External~~	~~No dependencies~~	~~--~~
FPT_STM.1	No dependencies	--
FPT_TST_EXT.1	No dependencies	--
FPT_PHP.1	No dependencies	--
FPT_PHP.3	No dependencies	--
FPT_FLS.1	No dependencies	--
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FMT_SMF.1	No dependencies	--
FMT_MTD.1/Unblock	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1/AuditLog	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MSA.1/GenKeys	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
FMT_MSA.1/AKeys	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1/KeyUsage

Requirement	Dependencies	Fulfilled by
	FMT_SMR.1 FMT_SMF.1	FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1
FMT_MSA.3/Keys	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/GenKeys, FMT_MSA.1/AKeys FMT_SMR.1
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1 FIA_UID.1
FAU_STG.2	FAU_GEN.1	FAU_GEN.1

Table 9 SFR Dependencies Rationale

8.2.4 Rationale for SARs

The assurance level for this Security Target is **EAL4 augmented with AVA_VAN.5 and ALC_FLR.2**.

EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialised processes and practices. It is considered to be the highest level that could be applied to an existing product line without undue expense and complexity. As such, EAL4 is appropriate for commercial products that can be applied to moderate to high security functions.

The TOE described in this Security Target is just such a product.

Augmentation results from:

- the selection of **AVA_VAN.5**. All the dependencies of AVA_VAN.5 are satisfied by other assurance components in the EAL4 assurance package,
- the selection of **ALC_FLR.2**, which has no dependencies.

AVA_VAN.5 Advanced methodical vulnerability analysis

The TOE generates uses and manages the highly sensitive data in the form of secret keys, at least some of which may be used as signature creation data. The protection of these keys and associated security of their attributes and use in cryptographic operations can only be ensured by the TOE itself. While the TOE environment is intended to protect against physical attacks, a high level of protection against logical attacks (especially those that might be carried out remotely) is also necessary, and is therefore addressed by augmenting vulnerability analysis to deal with High attack potential.

8.2.5 ALC_FLR.2 Flaw reporting procedures

Provides assurance that nCipher Security have well-defined processes and a proactive posture to act upon security vulnerabilities found in the field.

9 Bibliography

[CC1]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1 Revision 5, April 2017, CCMB-2017-04-001
[CC2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements, Version 3.1 Revision 5, April 2017, CCMB-2017-04-002
[CC3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance requirements, Version 3.1 Revision 5, April 2017, CCMB-2017-04-003
[AIS 31]	A proposal for: Functionality classes for random number generators, Version 2.0, 18 September 2011
[ISO 19790]	ISO/IEC 19790:2012 Information technology – Security techniques – Security requirements for cryptographic modules
[SP 800-90A]	NIST Special Publication 800-90A Rev. 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015
[CEN EN 419221-5]	CEN, EN 419221-5:2018, Protection Profiles for TSP Cryptographic Modules - Part 5, Cryptographic Module for Trust Services, Version 1.0
[TUAK]	ETSI TS 135 231
[Milenage]	ETSI TS 135 206
[FIPS 186-4]	Digital Signature Standard (DSS), July 2013
[FIPS 197]	Advanced Encryption Standard (AES), November 2001
[SP 800-38A]	Recommendation for Block Cipher Modes of Operation: Methods and Techniques, December 2001
[SP 800-38B]	Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication, May 2005 (Updated 10/6/2016)
[SP 800-38C]	Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, May 2004 (Updated 7/20/2007)

[SP 800-38D]	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007
[SP 800-67]	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Rev. 1, January 2012
[FIPS 180-4]	Secure Hash Standard (SHS), August 2015
[FIPS 198-1]	The Keyed-Hash Message Authentication Code (HMAC), July 2008
[SP 800-56A]	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, Rev. 2, May 2013
[SP 800-38F]	Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, December 2012
[SP 800-108]	Recommendation for Key Derivation Using Pseudorandom Functions (Revised), October 2009
[PKCS#1]	RFC 8017 PKCS #1 v2.2
[FIPS 140-3]	Security Requirements for Cryptographic Modules, March 22nd, 2019

10 Terminology

For the purposes of this document, the acronyms, terms and definitions given in [CEN EN 419221-5] apply.

Common Criteria terms and definitions are given in [CC1].

Additional terms defined for the purposes of this document are listed below.

Term	Definition
ACS	Administrator Card Set - a set of smart cards used to control access to Administration functions.
OCS	Operator Card Set – a set of smart cards used to control access to keys.
Softcard	A logical token that is protected by a passphrase.
Hardserver	The nShield server software running on the nShield Connect XC or host server/PC in which the TOE is installed. It performs the following functions: • Command and reply translation between the clients and the HSM, • Forward log entries to a Syslog server for persistent storage.
Impath	Inter-module path. An nCipher proprietary secure protocol between two Hardserver instances.
Key blob	Key blobs (also known as Application Key Tokens) provide a mechanism for securely storing a key and ACL on insecure media. They provide both confidentiality and integrity. Key blobs can be stored externally of the TOE or internally in non-volatile memory.
Logical token	A logical token is a symmetric key used exclusively for the purpose of protecting other keys. They can be split into Shares using a quorum system based on Shamir's Secret Sharing algorithm which allows reassembly of a logical token using any *k* of a total of *n* shares (these values being chosen when the logical token is created). Each Share is stored encrypted on a smartcard or softcard, protected with a passphrase.
Token protected key	A key stored in a Key blob which is protected by a Logical Token and the TOE's Module key. The Logical Token needs to be loaded from a Softcard or an OCS quorum to unlock the use of this key.
Certifier key	A key that is required to unlock the use of a Certifier protected key.
Certifier protected key	A key stored in a Key blob which is protected by the TOE's Module key. The ACL of the key requires a Certifier key to be loaded in the TOE to unlock the use of this key.

Term	Definition
ACL	Access Control List. An ordered list of permission groups, which consists of one or more actions, and optionally restrictions.
Module key	Symmetric key stored inside the TOE which is used for Key Blob protection.
CodeSafe application	An embedded application running in the protected environment of the HSM. The CodeSafe application is sandboxed and does not have access to key material loaded into the HSM except through the same APIs, satisfying the same access controls, as applications that call the HSM from the host side.
NFKM	A Security World API that can be used manage keys, cardsets and softcards
qSCD	Qualified Signature (or Seal) Creation Device
TVD	Trusted Verification Device, a card reader for secure remote presentation of smartcards to the HSM.

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