

nShield Security World

nShield Security World v13.9.9 Security Manual

09 November 2026

© Entrust Corporation. All rights reserved.

Table of Contents

1. Introduction . . . 1
    1.1. Who should read this document? . . . 1
    1.2. Products covered by this manual . . . 1
    1.3. Product security objective . . . 1
    1.4. Product selection . . . 2
    1.5. Security manual authority and scope . . . 2
    1.6. Related documents . . . 3
    1.7. Reference documents . . . 3
2. Supply and Transportation . . . 4
    2.1. Trusted delivery . . . 4
    2.2. Software and firmware integrity . . . 4
    2.3. Tamper inspection . . . 4
3. Environment . . . 6
    3.1. HSM function and architecture . . . 6
    3.2. HSM environment controls . . . 7
4. Commissioning . . . 9
    4.1. Preparation . . . 9
    4.2. Installation . . . 9
    4.3. Hardware . . . 10
        4.3.1. Trusted Verification Device . . . 10
        4.3.2. Mode switch and jumper switches (nShield Solo only) . . . 10
    4.4. Network configuration . . . 11
        4.4.1. Firewall settings . . . 11
        4.4.2. Simple Network Management Protocol (SNMP) . . . 11
    4.5. Date and Time . . . 12
        4.5.1. Set the nShield Solo and nShield Connect Real-Time Clock (RTC) . . . 12
        4.5.2. Set the nShield Connect date and time . . . 13
        4.5.3. Set the Host date and time (nShield Solo only) . . . 13
    4.6. nShield Connect and client configuration . . . 13
        4.6.1. Configuring the Ethernet interfaces - IPv4 and IPv6 . . . 14
        4.6.2. Optionally configure the Connect's hardserver interfaces . . . 14
        4.6.3. Impath resilience . . . 14
        4.6.4. Configuring the RFS . . . 15
        4.6.5. Remote configuration . . . 16
        4.6.6. Configuring the nShield Connect to allow a crypto client . . . 17
        4.6.7. Configuring a client to communicate with an nShield Connect . . . 17
        4.6.8. Configuring a client to communicate through an nToken . . . 18

4.6.9. Configuring the serial console . . . 18
4.7. Logging and debugging . . . 19
4.7.1. Set up logging . . . 20
4.7.2. Audit Log . . . 21
4.7.3. nShield Connect tamper log . . . 21
4.7.4. nShield Connect system and hardserver logs . . . 22
4.7.5. nToken, nShield Edge and nShield Solo hardserver logs . . . 22
4.7.6. Hardserver and system log environment variables . . . 22
4.7.7. Debugging options . . . 23
4.8. Configure access control . . . 23
4.8.1. Security World key protection options . . . 23
4.9. Configure Security World . . . 23
4.9.1. Security World options . . . 23
4.9.2. Application interfaces . . . 24
4.9.3. Nonvolatile memory (NVRAM) options . . . 24
4.9.4. Loading a Security World after firmware update . . . 25
4.10. Remote services . . . 25
4.10.1. Remote Administration Service (RAS) . . . 25
4.10.2. Remote Operator . . . 26
4.11. SEE . . . 26
4.11.1. Security World SEE options . . . 26
4.11.2. RTC options . . . 27
4.12. Set up SSH communications between host and nShield 5 and later modules . . . 27
5. Access Control . . . 28
5.1. Security World access control architecture . . . 28
5.1.1. User groups and users to install Security World software . . . 28
5.1.2. Security World access control . . . 28
5.1.3. Application key access control . . . 29
5.2. Security World access control guidance . . . 30
5.2.1. Administrator Card Set (ACS) protection . . . 30
5.2.2. Module protection . . . 33
5.2.3. Softcard protection . . . 33
5.2.4. Logical token passphrase guidance . . . 34
5.2.5. OCS protection . . . 35
5.3. Application keys . . . 38
5.3.1. ACL restrictions for key wrapping/encapsulation keys . . . 38
5.4. nShield Connect front panel . . . 38
5.5. Configuring remote administration access to nShield Connect . . . 39
5.6. Role holder lifecycle guidance . . . 39

5.6.1. Roles . . . 39
5.6.2. Access rights withdrawn . . . 41
5.6.3. Dos and don'ts for access control mechanisms . . . 41
6. Operation . . . 43
6.1. Patching policy . . . 43
6.2. Set the RTC time . . . 43
6.3. Operator Card Set (OCS) quorum configurations . . . 43
6.3.1. Share keys across multiple HSMs . . . 43
6.3.2. Share key between users . . . 44
6.4. NVRAM key storage . . . 45
6.5. Config push feature . . . 45
6.6. Security World replacement options . . . 45
6.7. Host platform and client applications . . . 45
6.8. Preload utility . . . 47
6.9. Discarding keys . . . 47
6.10. Erasing a module from a Security World . . . 47
6.11. Replacing an OCS . . . 47
6.12. Replacing the ACS . . . 48
6.13. Firmware upgrade . . . 48
6.13.1. Setting the minimum VSN after a firmware upgrade . . . 48
6.14. Enabling and disabling remote upgrade . . . 49
6.15. Migrating keys to a v3 Security World . . . 49
6.16. Untrusted input validation . . . 50
7. Key Management . . . 51
7.1. Key management schema . . . 51
7.2. Security World security strengths . . . 51
7.2.1. KML type and security strength . . . 52
7.3. Application keys algorithms and key sizes . . . 53
7.4. Cryptoperiods . . . 54
7.5. Generating random numbers and keys . . . 55
7.6. Key backup . . . 55
7.7. Key import . . . 55
7.8. Key separation . . . 55
7.8.1. Single purpose keys . . . 55
7.8.2. SSH secure channel keys . . . 56
7.9. nShield JCA/JCE CSP . . . 56
7.9.1. Installing the nShield JCA/JCE CSP . . . 56
7.10. nShield PKCS #11 library . . . 56
7.10.1. Symmetric encryption . . . 56

7.10.2. PKCS #11 library with Security Assurance Mechanism . . . 57
7.10.3. nShield PKCS #11 library environment variables . . . 57
8. Physical Security . . . 58
8.1. nShield Edge physical security controls . . . 58
8.2. nShield Solo+ physical security controls . . . 58
8.3. nShield Solo SoloXC and nShield 5s physical security controls . . . 59
8.4. nShield Connect physical security controls . . . 60
8.4.1. Tamper event . . . 60
8.5. nShield card readers . . . 64
8.6. Tamper inspection . . . 64
9. Audit . . . 65
9.1. HSM and card reader location . . . 65
9.1.1. Physical inspection . . . 65
9.2. ACS and OCS . . . 65
9.3. Logs . . . 66
9.4. HSM Audit logging . . . 66
9.5. Audit Logging time . . . 67
9.6. nShield 5 Signed System Logs . . . 67
10. Support and Maintenance . . . 68
10.1. Security advisories . . . 68
10.2. Application and Operating System patching . . . 68
10.3. Connect fan tray module and PSU maintenance . . . 68
10.4. Solo XC fan and battery maintenance . . . 69
10.5. Maintenance mode . . . 69
10.6. Troubleshooting . . . 70
10.6.1. Debugging information for Java . . . 70
10.6.2. Contacting Entrust nShield Support . . . 70
11. Security Incident and Response . . . 71
11.1. Security incident monitoring . . . 71
11.2. Security incident management . . . 71
11.3. Security incident impact and response . . . 72
11.3.1. Compromised Key or Secret: A brute force attack on blobbed key outside of module . . . 72
11.3.2. Compromised Key or Secret: Attacker has subverted memory of HSM . . . 72
11.3.3. Compromised Key or Secret: passphrase for softcard is compromised . . . 73
11.3.4. Compromised Key or Secret: A quorum of OCS cards is compromised . . . 73
11.3.5. Compromised Key or Secret: A quorum of ACS cards is compromised . . . 74
11.3.6. Compromised Key or Secret: Soft KNETI . . . 75
11.3.7. Compromised Key or Secret: nToken KNETI . . . 75

11.3.8. Compromised Key or Secret: nShield Connect KNETI . . . 76
11.3.9. Compromised Key or Secret: Soft KNETI . . . 76
11.3.10. Compromised Key or Secret: nToken KNETI . . . 77
11.3.11. Compromised Key or Secret: Imported application keys . . . 78
11.4. Deleting a Security World . . . 78
11.5. Module failure . . . 79
11.6. Tamper incident . . . 79
12. Decommission and Disposal . . . 80
12.1. nShield Connect and nShield Solo . . . 80
12.1.1. Recycling and disposal information . . . 81
12.1.2. Security World . . . 81
13. Abbreviations . . . 82

1. Introduction

Good security practice requires procedural as well as technical measures to provide a comprehensive security environment for the protection of your cryptographic keys and data.

This guide provides advice to you on the secure operation of the product. It identifies procedural measures that should be deployed to support the secure operation of the nShield. The guidance should be used in the development of your Security Operating Procedures for your systems incorporating the nShield.

1.1. Who should read this document?

The guide should be used by the following people:

- Those responsible for the security policy and procedures for your systems incorporating the nShield
- Those responsible for commissioning the nShield
- Those responsible for administering the nShield
- Those responsible for auditing the nShield.

1.2. Products covered by this manual

The guide covers the following product variants:

- nShield Edge
- nShield Solo+
- nShield Solo XC
- nShield 5s
- nShield Connect+
- nShield Connect XC
- nShield 5c

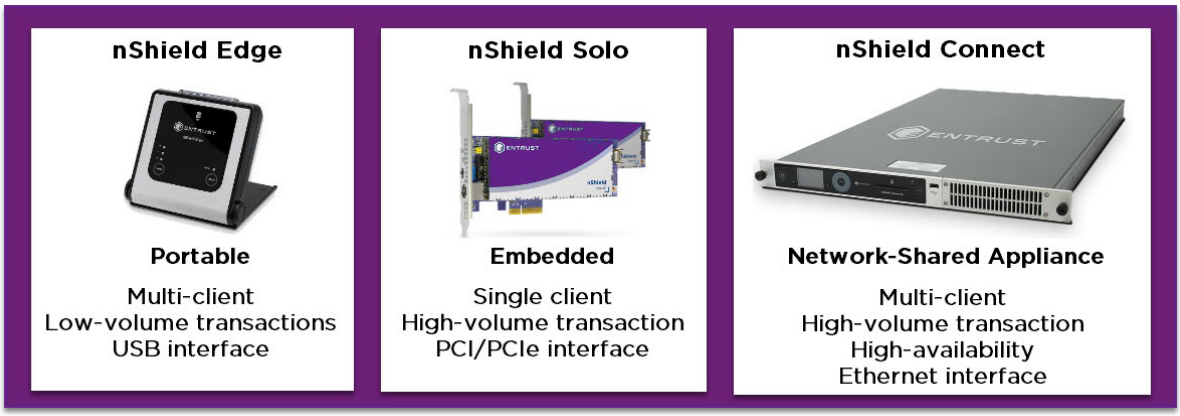


In this manual, guidance given for nShield Solo applies to the Solo+, Solo XC, and nShield 5s product variants. Similarly, guidance given for nShield Connect applies to the Connect+, Connect XC, and nShield 5c product variants.

1.3. Product security objective

The nShield range of products provide protection against technical and physical attacks on keys used to protect your data in use, in motion and at rest. This provides confidentiality, integrity and availability* of user data up to FIPS 140 Level 3 and Common Criteria version 3.1 revision 5 EAL 4+ (platform and version dependent) when deployed in accordance with the technical and procedural controls identified in the HSM and Security World product doc umentation and here in the *Security Manual*.

*Some availability threats can be mitigated by hosting a Security World across multiple Hardware Security Modules (HSMs).



1.4. Product selection

As part of the security product selection process, you must determine that the functionality delivered by any candidate product meets your requirements.



In this manual the terms module and HSM are both used to generically describe the nShield range of products.

1.5. Security manual authority and scope

The guide is advisory and its scope is limited to identifying good procedural practices for the secure operation of the product within your environment.

If there is any contradiction between the guidance that occurs in this manual and that found in the HSM and Security World product documentation, then the guidance found here takes precedence.

The scope of this manual is limited to security procedural guidance. The HSM and Security World product documentation provide guidance on how to implement the controls dis- cussed in this Manual.

1.6. Related documents

- nShield v13.9.9 Hardware Install and Setup Guides
- nShield v13.9.9 HSM User Guide
- nShield Security World Software v13.9.9 Installation Guide
- nShield Security World v13.9.9 Management Guide
- nShield v13.9.9 Utilities Reference
- nShield Security World v13.9.9 Key Management Guide
- CodeSafe 5 v13.9.9 Developer Guide
- CodeSafe v13.9.9 Developer Guide
- *nShield Connect Physical Security Checklist*

1.7. Reference documents

- Ecrypt-CSA recommendations
- NIST SP 800-57 Part 1 Revision 5
- NIST SP800-131A Revision 1
- Net-SNMP
- NTP Vulnerabilities.

2. Supply and Transportation

2.1. Trusted delivery

To help assure the integrity of the product during delivery a trusted courier service should be used that provides traceable delivery progress reporting and requires signed acceptance of delivery. Inspect the packaging for signs of tampering, for example, packing tape appears to have been removed or cut and then resealed. If tamper is detected, quarantine the package and notify your Security Officer in line with your Security Incident and Response procedures. Similar methods must be deployed by you for any further transportation of the product during its lifetime. If you utilize a protective marking scheme, the relevant protective marking must be deployed during transportation to provide the required level of integrity.

2.2. Software and firmware integrity

Entrust signs each nShield ISO image, including the Security World Software, CodeSafe, Remote Administration Client and firmware images, with its OpenPGP release key. One signature covers the image as a whole, and a second covers a manifest of the SHA-256 digests of the files on the image. The manifest and its signature are on the media, so you can verify the files on the physical media, on copies taken from the media, or in a downloaded image.

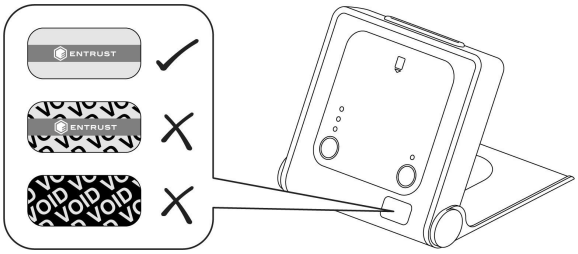
Verify the signatures before you install software or load firmware from the media, as described in Verify the installation media. The fingerprint of the release key is published in that guide. Compare the key against that fingerprint, not against the copy of the key on the media, because anyone able to alter the media could also replace the key on it.

If verification fails, do not install software or load firmware from the media. Quarantine the media and notify your Security Officer in line with your Security Incident and Response procedures.

2.3. Tamper inspection

Upon receipt of the nShield HSM, it must be inspected for signs of tamper:

- nShield Edge: Inspect the USB cable and the nShield Edge before use. The inspection should cover the cable for signs of tampering, the fascia for signs of disfigurement and specifically the holographic tamper label in the tamper window shown below for the appearance of **VOID**.



The nShield Edge Developer Edition does not have a hologram and tamper window.

- nShield Solo: Examine the epoxy resin security coating (after removing the metal lid on nShield Solo XC) of the module for obvious signs of damage.
- Smart card reader: Examine the smartcard reader for signs of tamper and ensure it is directly plugged into the module or into the port provided by any appliance in which the module is integrated and the cable has not been tampered with.
- nShield Connect: The *nShield Connect Physical Security Checklist*, provided in the box with an nShield Connect and available in the document folder on the installation media provides details of the physical security checks required. Further guidance on physical security checks can be found in Physical Security. Review the nShield Connect's tamper log for tamper alerts.
- See Physical Security for further guidance on the management of physical mechanisms provided to protect the product.

If a tamper is suspected then the unit must be quarantined to investigate the incident. The unit must not be deployed on the live system until its integrity can be verified. See the Security Incident and Response for further guidance.

3. Environment

3.1. HSM function and architecture

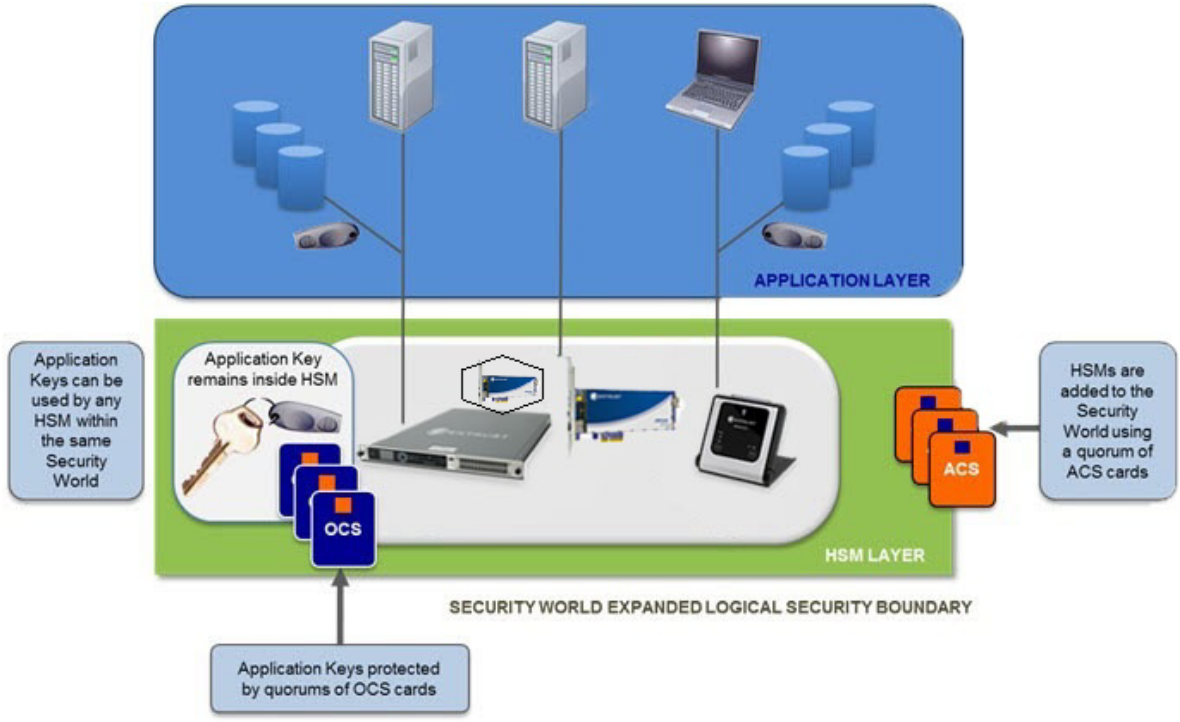
The nShield HSMs perform encryption, digital signing and key management on behalf of an extensive range of commercial and custom-built applications including Public Key Infrastructures (PKIs), identity management systems, application-level encryption and tokenization, SSL/TLS, and code signing.

nShield HSMs provide a hardened, tamper-resistant environment for secure cryptographic processing, key generation and protection and key, data and application encryption. They are available in three form factors to support a variety of deployment scenarios:

1. nShield Solo and 5s local FIPS 140 certified PCIe cards
2. nShield Connect and 5c network-attached appliances, that contain the nShield Solo or 5s FIPS 140 certified PCIe card along with surrounding networking and administration support facilities
3. nShield Edge FIPS 140 certified USB device

All nShield HSMs integrate with the nShield Security World architecture. This supports combinations of different nShield HSM models to build a unified ecosystem that delivers scalability, seamless failover and load balancing. The nShield Security World architecture supports a specialized key management framework that spans the nShield HSM range.

nShield HSMs all define the physical FIPS-certified security boundary or HSM Layer within which Application Keys, Control Keys and Infrastructure Keys are protected. Using quorums of Administrator Card Set (ACS) cards, Infrastructure Keys can be securely backed up and shared across multiple HSMs. When this is performed, HSMs that share the same Infrastructure Keys develop a common Security World that provides an expanded logical security boundary that extends beyond the physical HSM Layer and overlaps into the enterprise IT environment or Application Layer. The abstraction of Application Keys into Application Key Tokens enables these tokens to be stored outside the physical HSM and within the corporate IT environment.



The Security World technology makes sure that keys remain secure throughout their life cycle. Every key in the Security World is always protected by another key, even during recovery and replacement operations. Because the Security World is built around nShield key-management modules, keys are only ever available in plain text on secure hardware.

nShield Connect and nShield Solo HSMs also provide a secure environment for running sen sitive applications. The CodeSafe option lets you execute code within nShield boundaries, protecting your applications and the data they process. The CodeSafe area occurs outside of the module area that is FIPS 140 Level 2 and 3 approved.

The nShield HSM is used to protect sensitive keys, data and optionally applications. It can only operate securely if its environment provides the procedural security that it requires and if its security enforcing functions are utilized appropriately.

When configured correctly the nShield HSM provides encryption, digital signing and key management services in support of confidentiality and integrity requirements for your data. The nShield HSM is not designed to be completely resistant to denial of service attacks - these can be addressed by other aspects of the system design if warranted by the threat and impact assessments.

3.2. HSM environment controls

You must exercise due diligence to ensure that the environment within which the nShield HSMs are deployed is configured properly and is regularly examined as part of a comprehen

sive risk mitigation program to assess both logical and physical threats. The client's application and its environment must be protected from malware as they access the HSMs cryptographic services. Adequate logical and physical controls should be in place to ensure that malware is detected.

Your Security Procedures should identify the measures required to ensure the physical security (and counter any threats of theft or attack) of the nShield HSM, and associated host/client/Remote File System (RFS) platforms, backup data, Security Information and Event Management (SIEM) collectors and card readers.

Access to the nShield HSM, and associated host/client/RFS platforms, backup data, SIEM collectors and card readers secure areas must:

- Only be provided to authorized individuals
- Only be provided when necessary
- Subject to audit control.

The nShield HSM and any card readers integrate with your infrastructure/network. Therefore, any Security Policy requirements for the infrastructure/network must cover the nShield HSM and card readers as well when operating within that infrastructure/network.

The nShield HSM must be subject to protection against excessive processing demands.

The nShield HSM and any card readers must be shielded against electromagnetic emission detection to prevent potential side-channel attacks through emissions analysis, if this is considered a threat in the deployed environment.

Temperature range restrictions apply to the nShield Solo and the nShield Connect when in operation. The HSMs must be located in well ventilated locations (hosts or comms racks).

Voltage range restrictions apply to the nShield Solo XC, the nShield Connect XC, and the nShield 5c when in operation. The HSMs must be protected with surge protection equipment.

To keep track of the nShield HSM and any card readers in your environment and aid any investigation in the event of loss, an asset id should be assigned to the product and a record of the nShield HSM and any card readers description, serial number and location be entered against the asset id in an asset register.

4. Commissioning

The commissioning process covers installing and configuring an nShield HSM for live operation. The commissioning process must be conducted by authorized individuals in a secure environment as described in HSM environment controls.

4.1. Preparation

Prior to commissioning:

- Perform a threat analysis of your deployment environment or use an existing threat analysis.
- Review the guidance in this *Security Manual* and determine the procedures and configuration required for your systems to meet the threats identified in your threat analysis. The requirements should be identified in a Security Policy so that all users understand the controls that are necessary to operate the system securely.
- Any host operating systems should be a current, supported version with the latest patches applied.
- It is recommended that anti-virus software is installed on the host system and maintained with automatic updates.

4.2. Installation

Entrust supplies standard component bundles that contain many of the necessary components for your installation and, in addition, individual components for use with supported applications. To be sure that all component dependencies are satisfied, you can install either:

- All the software components supplied
- Only the software components you require.

During the installation process, you will be asked to choose which bundles and components to install.

Your choice depends on a number of considerations, including:

- The types of application that are to use the module
- The amount of disk space available for the installation
- Your policy on installing software. While it may be simpler to choose all software components, you may have a policy of not installing any software that is not required to

reduce the threat level for vulnerabilities arising in software (especially services, even if unused).

Verify the installation media before you install software from it. See Software and firmware integrity.

4.3. Hardware

4.3.1. Trusted Verification Device

For use with the Remote Administration Client, Entrust supplies and strongly recommends the use of the nShield Trusted Verification Device (TVD). This specialized smart card reader allows the card holder to securely confirm the Electronic Serial Number (ESN) of the HSM to which they want to connect, using the TVD display.



Only use a TVD that has been obtained via a trusted supply chain.

The list of valid ESNs must be provided out-of-band for manual verification of the nShield HSM target. This list must be kept securely and made available only to authorized cardholders.

TVD users must be made aware that:

- Only the ESN of the HSM that was selected in the Remote Administration Client should be displayed.
- There will only be a single confirmation requested for setting up communication with that HSM.
- Any other text shown in the TVD display, or any other ESN that appears, regardless of pretext given in any user interface, is illegitimate and the card-loading should be cancelled if such text is present.

4.3.2. Mode switch and jumper switches (nShield Solo only)

The mode switch on the back panel controls the mode of the module. See Checking and changing the mode on nShield Solo and nShield Solo XC HSMs for more information about checking and changing the mode of an HSM.

You can set the physical mode override jumper switch on the circuit board of the nShield Solo to the **ON** position, to prevent accidental operation of the Mode switch. If this override jumper switch is on, the nShield Solo ignores the position of the Mode switch.



You can set the remote mode override jumper switch on the circuit board of the nShield Solo to the **ON** position to prevent mode change using the `nopclearfail` command. This should be done if, for example, a threat analysis determines that it may be possible for a remote malicious or negligent user to interrupt operational service. In this instance the security policies of your organization should require that the physical mode switch must be used to authorize mode changes. For example a trusted role holder has to be locally present to authorize the change.

4.4. Network configuration

4.4.1. Firewall settings

When setting up your firewall, you should make sure that the port settings are compatible with the HSMs and allow access to the system components you are using. See Firewall settings for default port numbers. Only open up the ports you require and limit the IP addresses supported on those ports to the ones required.

4.4.2. Simple Network Management Protocol (SNMP)

In the development of the nShield SNMP agent, we have used open-source tools that are part of the NET-SNMP project. More information on SNMP in general, and the data structures used to support SNMP installations, is available from the NET-SNMP project Web site: https://www.net-snmp.org/.

This site includes some support information and offers access to discussion email lists. You can use the discussion lists to monitor subjects that might affect the operation or security of the nShield SNMP agent or command-line utilities.

You should discuss any enquiries arising from information on the NET-SNMP Web site with Entrust nShield Support before posting potentially sensitive information to the NET-SNMP Web site.

The nShield SNMP Agent allows other computers on the network to connect to it and make requests for information. The nShield agent is based on the NET-SNMP code base, which has been tested but not fully reviewed by Entrust. We strongly recommend that you deploy the nShield SNMP agent only on a private network or a network protected from the global Internet by an appropriate firewall. The agent runs with reduced privileges on Windows and Linux in order to mitigate potential risks in the third party code.

The default nShield SNMP installation allows read-only access to the Management Informa-

tion Base (MIB) with any community string. There is no default write access to any part of the MIB. The few Object Identifiers (OIDs) in the nShield MIB that are writable (should write access be enabled) specify ephemerally how the SNMP agent presents certain information. No software or HSM configuration changes can be made through the SNMP agent.

Every effort has been taken to ensure the confidentiality of cryptographic keys even when the SNMP agent is enabled. The SNMP agent runs as a client application of the HSM, and all the security controls provided and enforced by the HSM are in effect. In particular, the nShield module is designed to prevent the theft of keys even if the security of the host system is compromised, provided that the Administrator Cards are used only with trusted hosts.

We strongly advise that you set up suitable access controls in the `snmpd.conf` file rather than using the default settings in production. It is also recommended that a firewall is used to restrict access to the agent to legitimate client machines.

When configuring the nShield SNMP agent, make sure that you protect the configuration files if you are storing sensitive information in them.

On Linux systems, the SNMP agent will run automatically if the ncsnmp package is installed, once the install script has been run. Therefore configuration should be done before installation in a production environment to ensure the correct settings are in place from the outset. On Windows systems, the SNMP agent runs only after enabling the service with `snmpd -register`, so configuration changes should be applied before this step.

4.5. Date and Time

The following sections provide procedural guidance about securely using date and time functions. Please see the HSM User Guide for information on how to operate these functions.

4.5.1. Set the nShield Solo and nShield Connect Real-Time Clock (RTC)

Set the RTC using an accurate trusted local time source as part of the commissioning process. This must be set as early in the commissioning process as possible. The correct time must be set to support hardserver and audit logging.



The backup battery for the RTC on the nShield Solo and nShield Solo+ will only last for two weeks when the module is not powered. The RTC must be reset on power up in such circumstances, that is, when RTC battery exhaustion occurs. See rtc for more information on resetting the

clock.

4.5.2. Set the nShield Connect date and time

Set the nShield Connect date and time using an accurate trusted local time source as part of the commissioning process. This must be set as early in the commissioning process as possible. The correct time must be set to support system, hardserver and tamper logging.

The nShield Connect supports a Network Time Protocol (NTP) client which if activated will synchronize the nShield Connect time to an NTP enabled time source.



NTP has featured many security vulnerabilities: https://www.cvedetails.com/vulnerability-list/vendor id-2153/NTP.html. The activation of NTP within the nShield Connect can increase the threats the nShield Connect is exposed to. Due to the nature of NTP design not all threats can be mitigated. NTP should only be used if your risk analysis identifies suitable controls to mitigate the impact of its operation. This could include:

- Using only NTP servers that are under the control of the customer, that is, within the customer's enterprise
- Using multiple NTP sources to mitigate an attack on a single NTP source or failure of that source. NTP can provide an accurate time source through consensus with multiple input servers. It can also identify which available time servers are inaccurate

To further mitigate attacks on NTP the synchronized time should be compared against the Solo RTC time (including within the Connect) at regular intervals to ensure that a similar time, within a margin of error, is reported. Significant discrepancies should be investigated.

To identify NTP server failures or attacks, NTP servers should be monitored for availability on the network and an alert generated if the NTP server is unavailable.

4.5.3. Set the Host date and time (nShield Solo only)

Set the host date and time using an accurate trusted local time source as part of the commissioning process. This must be set as early in the commissioning process as possible. The correct time must be set to support hardserver logging.

4.6. nShield Connect and client configuration

4.6.1. Configuring the Ethernet interfaces - IPv4 and IPv6

The nShield Connect has two Ethernet interfaces. Depending on the network configuration guidelines in the customer's security policy, separated networks can be supported by assigning one interface an internal IP address, and the other an external IP address. When these Ethernet interfaces do not need to be used separately, they can be bonded to provide either redundancy or higher throughput. Configure an Ethernet bond interface provides more information on configuring Ethernet bonding. If an Ethernet interface is not used, disable it.

4.6.2. Optionally configure the Connect's hardserver interfaces

By default, the hardserver listens on all of its interfaces. However, you can alter the hardserver settings. Altering the hardserver settings would prove necessary if, for example, you wanted to only connect one of the Ethernet interfaces to external hosts.

Ensure that you have configured the Ethernet interfaces on the HSM before attempting to configure the hardserver. The hardserver should not be permitted to listen on any interface that is connected to the public Internet, as this could expose the HSM to attack from network based attackers. The HSM should also be placed behind a firewall during commissioning in order to restrict network traffic to only legitimate clients and RFS.

4.6.3. Impath resilience

The `nethsm_settings` section in the client host hardserver config file defines settings for Impath resilience that are specific to the nShield Connect. Impath resilience is a session resumption feature which allows client hardservers to reconnect securely to their existing session with the Connect from a fresh TCP connection after a network outage. The session resumption requires both the usual mutual authentication as well as an additional authentication step to enable the client to access the previous session, that ties it to the same client hardserver instance. By default Impath resilience is turned on with a timeout of 1,800 seconds (30 minutes). This enables clients to reconnect in the event of network errors without needing to reload keys. An associated time-out can be configured to state when an Impath resilience session will expire after which all previously loaded objects must be reloaded. Your threat analysis of your environment, and knowledge of the reliability of your network, will determine if Impath resilience needs to be enabled, and what timeout should be set. For example, a five-minute Impath resilience timeout could give a reasonable trade-off between security and resilience to transient network issues.

4.6.4. Configuring the RFS

The nShield RFS (Remote File System) is a protocol for securely sharing access to files between machines.

The two primary uses for the RFS are as a

1. file store for the nShield Connect `config` file, logs and as a source for nShield Connect upgrade images, feature files, CodeSafe application files and (where the nShield Connect front panel is being used for Security World operations) world and cardset files. This usage is often referred to simply as "the RFS".
2. way to share Security World files (including encrypted key blobs) between peer machines, usually using the `rfs-sync` tool. This usage is also referred to as cooperating clients.

By default, this protocol runs on port 9004, which will need to be enabled in the machine's firewall for TCP communication. If a client machine is not being used as an RFS, it is recommended that port 9004 be disabled in the `config` file `[server_remotecomms]` section. It is also possible to restrict to listening only on a particular IP address or network interface using this configuration section.

4.6.4.1. nShield Connect RFS

You should regularly back up the entire contents of the RFS as it is required to restore an nShield Connect or its replacement, to the current state in the case of failure.

To setup, the RFS requires certain information about the nShield Connect:

- IP Address
- ESN
- The hash of the KNETI (referred to as HKNETI)

Even with a trusted network, it is recommended that the ESN and KNETI reported by `anonkneti` be checked independently using the nShield Connect front panel, or from the serial console. If the network is untrusted, obtaining the ESN and KNETI information directly from the nShield Connect front panel is essential. This information is then used in the `rfs-setup` command to create the RFS. Specifically the `--write-noauth` option should not be used with the `rfs-setup` command over insecure networks as this does not authenticate the RFS which could give rise to a masquerade attack.

It is strongly recommended that mutual secure authentication is enabled between the nShield Connect and the RFS, that is both that `rfs-setup` records the authentication information for the nShield Connect in the RFS `config` file as described above, but also that the

nShield Connect `config` file records the KNETI hash of the RFS in its configuration file. Authentication of the RFS by the nShield Connect can be specified when enrolling via the nShield Connect front panel, in which case the KNETI hash of the RFS machine that is displayed in the front panel must be checked against what is reported by `enquiry -m 0` or by `anonkneti -m 0 127.0.0.1` run locally on the RFS machine (for Software KNETI, change `-m 0` to `-m 1` etc. to query a module-protected KNETI). Authentication can also be specified in the serial console or in the nShield Connect `config` file. If the RFS is authenticated, by default it is also permitted as a privileged client, and as a config-push client, so it is the most convenient way to securely bootstrap all administration of the system.

Enabling 'secure authentication' to the RFS is described in Configuring the Remote File System (RFS) and Configuring the Remote File System (RFS) (both for network-attached HSMs).

4.6.4.2. Configuring Client Cooperation

If your machine needs to access to a peer RFS for sharing Security World files (that is, it's a cooperating client), it is strongly recommended to use each machine's Software Key KNETI (or an nToken-protected or local HSM-protected KNETI if the machine has one fitted) to specify secure authentication between RFS peers over an insecure network.

This means that the `rfs-setup --gang-client --write-noauth` option should **not** be used when configuring the RFS, and instead `rfs-setup` must be run with the HKNETI of the peer that is allowed to initiate connections to the machine where `rfs-setup` (or the ESN and the HKNETI) if the peer is connecting with an nToken-protected or local HSM-protected KNETI).

On the initiating end of the connection where `rfs-sync` is being run, similarly the HKNETI (and ESN if module-protected) of the peer must be specified in order to authenticate it by specifying the `--rfs-hkneti` (and optionally `--rfs-esn`) parameters to `rfs-sync --setup`.

These options are outlined in Client cooperation.

4.6.5. Remote configuration

The module (hardserver) configuration file can be used to enable:

- Remote reboot
- Remote mode changes
- Remote upgrade. If this functionality is not required then it must be disabled.

4.6.6. Configuring the nShield Connect to allow a crypto client

Before a crypto client of the nShield Connect can connect, it must first be added to the permitted client list in one of two ways:

1. nShield Connect front panel. It is recommended that the secure authentication option be selected, which will require that port 9004 be temporarily accessible on the client machine so that its authentication options (Software KNETI and if available module-protected KNETI) can be queried automatically. This port can then be disabled in the client after enrollment is completed unless it is also being used as an RFS machine. The KNETI hash of the selected authentication option will be displayed in the front panel, and must be confirmed against the KNETI hash reported locally on the client machine.
2. Push an updated `config` file to the nShield Connect with a new entry in the `[hs_clients]` section. The HKNETI (`keyhash` field) (and `esn` if necessary for module-protected KNETI key) should be filled in, along with specifying `clientperm` with the required access level.

Privileged connections can be restricted to have source port numbered less than 1024 (low ports) as a minor additional defence in depth measure as these ports are privileged on Linux.

It is strongly recommended that all clients (privileged or unprivileged) are configured to be authenticated with a KNETI network authentication key. This client key can be a software key, or an nToken or local HSM-protected key.

Even if the clients will be dynamically created (e.g. containers that are spun up as required), it is more secure to allow access to a KNETI key associated with that container, than to simply allow any users of the network to connect without authentication.

4.6.7. Configuring a client to communicate with an nShield Connect

A utility `nethsmenroll` is used to edit the configuration file of the client hardserver to add an nShield Connect with a given IP address or hostname. It is strongly recommended that the utility is used with the optional ESN and HKNETI parameters filled in. This content must be obtained from the nShield Connect's front panel, or from the nShield Connect serial console (using the `esn` and `kneti` commands). An alternative method for using `nethsmenroll` is to omit the ESN and HKNETI parameters. In this situation, `nethsmenroll` will attempt to recover ESN and HKNETI parameters from the nShield Connect and then prompts the client for confirmation that the values are correct. Confirmation is achieved by verifying the ESN and HKNETI displayed on the front panel of the nShield Connect (or in the serial console) are the same values as those reported by `nethsmenroll`. This step must be completed when enrolling clients over a network to verify that the client is communicating with the

valid, identified nShield Connect. Once the values are confirmed they are automatically written to the client-side configuration file and used for verification for all future connections. The HKNETI of an nShield Connect does not change in normal operation, although it does change in the event of factory reset. If the HKNETI ceases to match, resulting in existing client hardserver configurations refusing to connect to the nShield Connect, the cause of the change should be investigated, and if no satisfactory explanation is found, tampering should be suspected.

The `nethsmenroll` option `--no-hkneti-confirmation` to enroll without either supplying the ESN and HKNETI as parameters, nor confirming it interactively, should not be used in production and is present only for test automation purposes.

4.6.8. Configuring a client to communicate through an nToken

If an nToken is installed in a client, it can be used to both generate and protect a key that is then used for the Impath communication between the nShield Connect and the client. A dedicated hardware protected key is used at both ends of the Impath as a result. The nToken mitigates threats occurring in the client environment including vulnerabilities arising in generic software and operating systems.

When configuring an nShield Connect to use a client containing an nToken, you must obtain the nToken key hash from the client and then view the client's configuration from the front panel of the nShield Connect and verify that the nToken key hash displayed there matches the nToken key hash obtained from the client. This makes sure that the correct nToken will be enrolled.



If an nToken isn't installed, then it is recommended to use software-based authentication. If neither an nToken nor software-based authentication is available for secure authentication, the client connection relies solely on the nShield Connect validating the client's IP address against the configured value to authenticate the client. In this instance the 'credentials' used by the nShield Connect to authenticate the client are weaker than the 'credentials' used by the client to authenticate nShield Connect. The method for authenticating the client may be vulnerable to its IP address being spoofed by an attacker.

4.6.9. Configuring the serial console

The serial console on the nShield Connect is enabled by default and can be disabled from the front panel. Regardless of the serial console being enabled or disabled, factory resetting an nShield Connect will re-enable the serial console. Disable the serial console if serial con-

sole connectivity is not required to prevent unauthorized access attempts. This is important as the serial console is shipped with a known default password that could allow an unauthorized access if the serial console is enabled and the default password is not changed.

The Serial Console requires a serial cable connection to a serial port aggregator which in turn is connected to an Administrator console via a communication channel. The administrator must ensure that a secure channel is correctly setup between their console and an authenticated serial port aggregator. Physical access controls should be deployed to protect the following from unauthorized access attempts:

- Serial cable
- Serial port aggregator
- Administrator console

Logical access controls should be deployed to protect the following from unauthorized access attempts:

- Serial console
- Serial port aggregator
- Administrator console

All passwords should be protected from unauthorized access or viewing.

The username for accessing the serial console is cli and the default password is admin. On first login you will be prompted to change the password for the cli user. The minimum length of the new password is 5 characters. The functionality does not enforce strong passwords therefore these should be manually implemented – see the relevant password guidance contained in Access Control Chapter sections:

- Logical Token passphrase guidance
- Dos and don’ts for access control mechanisms

Equipment that supports the serial console connection must be maintained with the latest software and patches, such as the serial port aggregator or the administrator console.

4.7. Logging and debugging

The following sections provide procedural guidance about securely using logging and debugging functions. Please see the HSM User Guide for information on how to operate these functions.

4.7.1. Set up logging

Once the time has been set, your logging requirements should be identified and implemented.

Logs are available on nShield platforms:

Type of Log	Purpose and configuration	nToken, Edge	Solo+, Solo XC, nShield 5s	Connect+, Connect XC, nShield 5c
Audit logging	Operational and key usage activity Can only be enabled at Security World initialization	Yes	Yes	Yes
Hardserver log	Errors and troubleshooting Controlled through environment variables. Default is to log nothing. Level of logging can be set.	Yes	Yes	Yes
Security World log used by API libraries and base services	Errors and troubleshooting Controlled through environment variables. Default is to log nothing, unless this is overridden by any individual library. See Environment variables for more information of `NFLOG_*` variables. If any of the four logging variables are set, all unset variables are given default values. Level of logging can be set.	Yes	Yes	Yes
System log	System events on nShield Connect Always enabled	No	No	Yes
Tamper log	Tamper events on nShield Connect Always enabled	No	No	Yes
`Net-SNMP` daemon log	`Net-SNMP` daemon activity Starts with the SNMP service. The SNMP Agent contains a logfile switch to record warnings and messages.	Yes	Yes [1]	Yes [1]

[1] The `netsnmp` daemon log is produced by the client.

4.7.2. Audit Log

Only the Audit Log originates in the nShield Solo HSM (including the nShield Solo HSM installed in every nShield Connect). It uses a cryptographic mechanism to assure the integrity of the audit log distributed to third party SIEM Collectors.

- The SIEM Collectors should be located in a protected environment that limits physical access to the processing platform(s), on which the collector and validation applications are running, to authorized users.
- The availability of log messages delivered to the SIEM Collectors should be maintained through a set of controls including:
  - Configuring multiple SIEM Collectors for each HSM
  - Regular backups
  - Physical and logical access controls for accessing backups.
- A mechanism should be supplied to support the reliable delivery of logging messages to the SIEM Collectors.
- The network should be configured correctly to help prevent message corruption, congestion, forwarding loops and incorrect delivery.
- The Audit Logging Verification process provided by Entrust to support the authenticated supply of a Trusted Root to the customer should be followed. See Audit log verification for further information on Audit Logging Verification.
- Prior to shutting down the nShield Connect, a delay of at least 17 minutes should be made after the final log messages has been dispatched to the SIEM to ensure that the outstanding integrity verification message for those log messages is dispatched. This requires that no further commands are entered that generate log messages during this period. The receipt of the verification message on the SIEM should be confirmed prior to HSM shutdown.
- The integrity verification messages allows missing or altered log messages to be detected. In the event of a power failure, or an SOS on the nShield Connect, the loss /manipulation of any log message received after the last integrity verifications message cannot be detected, and therefore these log messages cannot be trusted.

4.7.3. nShield Connect tamper log

The nShield Connect's Tamper Log is located within the nShield Connect and protected by the nShield Connect's tamper mechanisms. It cannot be erased using the Connect front panel or serial console.

The tamper log is an informational tool providing an indication of events in normal opera-

tion. It is not intended as a security control and cannot be relied on if there are reasons to believe that device has been tampered with (e.g. through examination of physical marks on the device).

It is essential that physical access controls are in place to prevent unauthorized access to the device to prevent tampering.

In addition to physical access controls, enabling and monitoring HSM audit logs (Security World audit logs enabled when creating the world, and additionally signed system logs in the case of the nShield 5s and 5c) are the effective measures to prevent and discover misuse rather than relying on the Connect tamper log.

4.7.4. nShield Connect system and hardserver logs

The nShield Connect's System and Hardserver Logs can be stored within the nShield Connect's tamper response boundary or pushed out to the RFS or a remote syslog server. If the logs are stored locally then they will be protected by the tamper response boundary but will be lost if the nShield Connect is rebooted. Logging stops when the file system is full. Logs stored on in the nShield Connect can be viewed using the Front Panel or fetched on demand using the `appliance-cli` client tool from a privileged client.

If logs are configured to be pushed to the RFS, they will be sent securely to the RFS endpoint (with mutual authentication, provided this is configured, the secrecy and integrity of messages are protected in transit). Logs may be configured to be pushed every minute to minimize the loss of data in the event of reboot or power loss.

Alternatively, the logs can be streamed to a remote syslog server over UDP. However, in this situation no authentication, secrecy or integrity mechanism is applied to the logs. This is only recommended on a trusted network (such as a private network between one of the Connect interfaces and host machines).

4.7.5. nToken, nShield Edge and nShield Solo hardserver logs

For nToken, nShield Edge and nShield Solo products the hardserver logs are stored in the associated host platform and do not have an integrity mechanism applied to the logs. There fore, physical, procedural and technical controls should be applied to the host platform envi ronment to protect the integrity of the logs.

4.7.6. Hardserver and system log environment variables

For Hardserver and System logging, environment variables can be applied to control the

amount of logging information. See Environment variables for more information. Your system management policy and security policy will determine the level of logging required.

4.7.7. Debugging options

Debugging information is available for hardware, application, Application Programming Interfaces (APIs) and operating systems. Unless required for development debugging for these sources should not be enabled (it is disabled by default).

Some debug information is provided in command line responses and therefore cannot be disabled.



SEE debugging is disabled by default, but you can choose whether to enable it for all users or whether to make it available only through use of an ACS.

Debugging can leak potentially useful information to an attacker so should not be enabled in production environments.

4.8. Configure access control

4.8.1. Security World key protection options

For guidance on access control mechanisms available to protect application keys in a Security World see Access Control.

4.9. Configure Security World

4.9.1. Security World options



The HSM must be in a secure and trusted environment before a Security World is created or loaded.

Decide what kind of Security World you need before you create it. Depending on the kind of Security World you need, you can choose different options at the time of creation. For convenience, Security World options can be divided into the following groups:

- Basic options, which must be configured for all Security Worlds. This includes environment variables.

- Recovery and replacement options, which must be configured if the Security World, keys, or passphrases are to be recoverable or replaceable. If you disable OCS and softcard replacement, you can never replace lost or damaged OCSs generated for that Security World. Therefore, you could never recover any keys protected by lost or damaged OCSs, even if the keys themselves were generated as recoverable (which is the default for key generation). Replacing OCSs and softcards requires authorization. To prevent the duplication of an OCS or a softcard without your knowledge, the recovery keys are protected by the ACS. However, there is always some extra risk attached to the storage of any key-recovery or OCS and softcard replacement data. An attacker with the ACS and a copy of the recovery and replacement data could re-create your Security World. If you have some keys that are especially important to protect, you may decide:
  - To issue a new key if you lose the OCS that protects the existing key
  - Turn off the recovery and replacement functions for the Security World or the recovery feature for a specific key.

  The recovery data for application keys is kept separate from the recovery data for the Security World key. The Security World always creates recovery data for the Security World key. It is only the recovery of application keys that is optional. See Access Control for more information and guidance on the options available.
- SEE options, which only need be configured if you are using CodeSafe
- Options relating to the replacement of an existing Security World with a new Security World.

Security World options are highly configurable at the time of creation but, so that they will remain secure, not afterwards. For this reason, we recommend that you familiarize yourself with Security World options, especially those required by your particular situation, before you begin to create a Security World.

4.9.2. Application interfaces

A Security World can protect keys for any applications correctly integrated with the Security World software. A wide range of application interfaces are supported. The application interfaces have many configuration settings. These should be reviewed to identify that only the required settings are activated and that all others are set to off.

4.9.3. Nonvolatile memory (NVRAM) options

Application keys can be stored in the module's NVRAM instead of in the `Key Management`

`Data` directory of the host computer. However, the space in NVRAM is limited and does not provide greater security than storing keys in the `Key Management Data` directory of the host. Therefore, it is recommended that encrypted key blobs are stored in the `Key Management Data` directory where space is limited only by the capacity of the host computer.

4.9.4. Loading a Security World after firmware update

Before initialising a Security World on an HSM after a firmware upgrade, it is recommended that the ACS card holder quorum check that the version of the firmware is correct. This will mitigate the possibility of rollback/downgrade attack.

4.10. Remote services

4.10.1. Remote Administration Service (RAS)

To use Remote Administration with nShield Connects, the RAS must be installed on a machine that is enrolled as a client of the nShield Connects.This may be the RFS machine.

To use Remote Administration with nShield Solo(s), the RAS must be installed on the host where the hardserver and nShield Solo(s) reside. If you have multiple nShield Solo hosts in a Security World, the RAS must be installed on each one. It will also be necessary to install the KLF2 warrants on the host machine if using Solo+ or Solo XC; this is not required for nShield 5s nor with nShield Connects where the KLF2 warrant is built-in.

The remote access solution that your organization normally uses, such as Secure Shell (SSH), remote PowerShell or a remote desktop application, is also required for running the nShield client applications that will use the remotely presented ACS or OCS cards.

A secure private communications channel, such as a Virtual Private Network (VPN), or an SSH tunnel (to TCP port 9005), must always be used for the connection between the Remote Administration Client (RAC) and the RAS if they are on separate computers.

You must use nShield Remote Administration Cards with Remote Administration. These are smart cards that are capable of negotiating mutually authenticated cryptographically secure connections with an HSM, using warrants as the root of trust.

Dynamic slot timeouts are available to define timeout values for expected smartcard respon siveness and round trip latency for all HSMs that are using the Remote Administration. Expected network delays need to be taken into account when setting this. The timeouts provide control over the period of time a smart card will remain responsive in the system that is experiencing unreasonable latency (where the system's non-performance could be a

consequence of an attack).

The `cardlist` file must be used to specify the serial numbers of remote cards that are allowed to be used by nShield client tools as an extra defence-in-depth facility to ensure only legitimate smartcards are presented.

If Remote Administration is not required then set the dynamic slots to `0` to disable Remote Administration.

4.10.2. Remote Operator

The Remote Operator feature enables the secure transmission of the contents of a smart card inserted into the slot of one attended module to another unattended module. To trans mit to a remote module, you must make sure that:

- The smart card is from a persistent OCS – see Access Control for guidance on persistent OCS.
- The attended and unattended modules are in the same Security World.

By default, modules are initialized into Security Worlds with remote card set reading enabled. Disable the Remote Operator feature if there is no requirement for it.

4.11. SEE

When an HSM is used to protect application keys, the keys are only available in unencrypted form when they are loaded into the HSM. However, traditionally, the code that uses the keys remains on the server. This means that the code is open to attack. It is possible that the code could be modified in such a way as to leak important information or compromise your business rules.

By implementing a solution with the SEE, you not only protect your cryptographic keys but also extend the physical security boundary to include your security critical code and data. Using the techniques of code signing and secure storage, the SEE enables you to maintain the confidentiality and integrity of application code and data and to bind them together so that only code in which you have confidence has access to confidential data.

If your threat analysis determines that the application code is vulnerable to attack, then it is recommended that you use the integrity and confidentiality protections that are provided by an SEE machine. See the *CodeSafe Developer Guide* for more details.

4.11.1. Security World SEE options

You must configure SEE options if you are using the nShield SEE. If you do not have SEE installed, the SEE options are not applicable.

4.11.2. RTC options

RTC options are relevant if you have purchased and installed the CodeSafe Developer kit. If so, by default, Security Worlds are created with access to RTC operations enabled. However, in FIPS Level 2 and Level 3 Security Worlds an option is available during Security World initialization to control access to RTC operations by means of an ACS. A threat analysis can determine if access to the RTC requires ACS authorization.

4.12. Set up SSH communications between host and nShield 5 and later modules

SSH secure channels protect communications between the host and nShield 5 and later HSMs. To allow mutual authentication of the endpoints, the SSH protocol uses separate key pairs in the host and the HSM. The functionality within the HSM is divided into different services that each use separate SSH channels. Before you can use these services, SSH client public keys must be installed on each service.

If allowed by your security policy, make a backup of your `sshadmin` private key. This means that you can re-establish communication with the HSM if any of the other installed service keys are erased or otherwise lost.

You must regard SSH keys that connect the host to the HSM as sensitive assets and protect them appropriately. On the HSM, the private keys are stored on flash memory. On the client, they are stored on the host and are protected by OS filesystem permissions where only root or local Administrators have write access, and only root, nfast user or permitted groups have read access depending on the type of service key. The client key files are stored in an encrypted format.

The default protection configuration and the available options to control how the client keys are protected are explained in detail in SSH Client Key Protection (nShield 5s HSMs). You must secure access to the client host machines, any backup files and any passphrases you optionally set on the keys, to prevent them from being used by an adversary to retrieve the client private keys stored within the files.

5. Access Control

An nShield HSM Security World can be configured and managed both remotely and locally using the supplied access control mechanisms.

5.1. Security World access control architecture

This section describes the access control options available within a Security World and the pros and cons of each option. This guidance is provided to help the customer to determine the right options for their threat environment.

5.1.1. User groups and users to install Security World software



On Linux and Windows, `root` or administrator privileges are required to install Security World software, and to configure access to local HSMs.

On Linux, run nShield utilities as a normal user if only unprivileged nCore commands and read-only access to keys is needed, as a user in the group `nfast` for privileged nCore commands and ability to modify kmdata, or in the group `nfastadmin` for access to `hsmadmin` and `csadmin` operations. Do not run production workloads as `root` or as the `nfast` user.

On Windows, ordinary local users can run unprivileged nCore commands and have read access to kmdata keys by default. Administrator privileges are required by default for privileged nCore commands and writing to the `Key Management Data` directory, but custom settings can be set in the `config` file for the groups allowed to make privileged and unprivileged connections, and either manual filesystem DACL changes or a non-default `Key Management Data` directory used to enable alternative access controls.

See Roles for further details.

5.1.2. Security World access control

All Security Worlds are protected by an ACS created at Security World initialization. The ACS is used to:

- Control access to Security World configuration

- Authorize recovery and replacement operations.

The ACS consists of a number of smart cards, N, of which a smaller or equal number, K, is required to authorize an action. The required number K is known as the quorum. The cards are distributed amongst authorized role holders so that a quorum of role holders are required to authorize the above operations.

See Administrator Card Set (ACS) protection for guidance on configuring and protecting the ACS.

5.1.3. Application key access control

The Security World and nShield HSM provide the facility for different levels of application key protection. There are three levels of application key protection:

- Module protection
  - The key is simply protected by the HSMs in the Security World
  - Any application on the host can load and use the key (but not export from HSM, unless ACLs allow).
- Softcard protection
  - The key is additionally protected by a passphrase
  - An application will prompt you for the passphrase before loading the key.
- OCS (token) protection.
  - The key is protected by a card set
  - Each card set consists of a number of smart cards, N, of which a smaller or equal number, K, is required to authorize an action. The required number K is known as the quorum.
  - The smartcards are distributed amongst authorized role holders so that a quorum of role holders are required to authorize loading of an application key.
  - The role holders must present the required cards to authorize the key loading.
  - The smartcard can be optionally protected with a passphrase. This can be set at any time.
  - The smartcards can be created in persistent or non-persistent (default) mode and with or without an associated time-out to provide different user options dependent on the importance of the application keys, and physical and logical security controls available in the environment.

Module-protected keys have no passphrase and are usable by any instance of the application for which they were created, provided that the application is running on a server fitted

with a HSM (or connected to an nShield Connect) which is initialized with the same Security World that was used to create these keys.

This level of protection is suitable for high-availability web servers that you want to recover immediately without intervention if the computer resets. However, the environment should be secure as the ability to use the application keys is dependent on any underlying access control provided by the associated operating systems hosting the application instances. See Module protection for guidance on the controls required to prevent unauthorized key access to module-protected keys.

The addition of a passphrase allows tighter control over key usage through explicit authorization. Controlling access to a key via a passphrase is achieved through creating a softcard. A softcard is a file containing a logical token that you cannot load without a passphrase. You must load the logical token to authorize the loading of any application key that is protected by the softcard. A softcard functions as a persistent 1/1 logical token. After a softcard's logical token is loaded, it remains valid for loading its keys until its key handle is destroyed. A single softcard can protect multiple application keys. See Softcard protection for guidance on the controls required to prevent unauthorized access to keys protected by a softcard.

The highest level of protection available for application keys is provided by OCS-protection. It is recommended that OCS cards additionally be passphrase-protected. When keys are protected by an OCS, an attacker would need to obtain access to a quorum of the OCS smartcards, plus any associated smartcard passphrases, to be able to load the logical token associated with the OCS. Only when the OCS's logical token has been loaded onto the HSM can the OCS-protected keys be loaded and used in operations. The OCS's logical token will remain loaded until its reauthorization timeout is reached, the last OCS smartcard is removed (for non-persistent tokens), or it is destroyed. A single OCS can protect multiple application keys. See OCS protection for guidance on the controls required to prevent unauthorized access to keys protected by a OCS.

All Security Worlds rely on you using the security features of your operating system to control the users who can access the Security World and, for example, write data to the host.

Your security policy (in response to a threat analysis) will determine the level of protection required for your application keys.

5.2. Security World access control guidance

5.2.1. Administrator Card Set (ACS) protection

There is always only one ACS in a Security World. When creating a Security World you can configure the following options:

- Whether to enable key recovery

  If you disable the key recovery option, you cannot replace lost or damaged OCSs and, therefore, cannot access keys that are protected by such cards. This feature cannot be enabled later without reinitializing your Security World and discarding all your existing keys.

- The number of administrator cards required to authorize key recovery
- Whether to enable passphrase recovery
- The number of administrator cards required to authorize passphrase recovery
- The number of administrator cards required to load this Security World onto a new HSM.
- The number of administrator cards required for various features.
- Whether to disable certain features
- Whether to specify a passphrase for a card (passphrases are strongly recommended)

Your threat analysis should determine which features and/or options are enabled and ACS quorums required to authorize those features. It is recommended that the quorum for the ACS is larger than the quorum required to activate any feature. This will ensure that the ACS quorum is only ever used when legitimately required. Additionally the quorum size for Key and passphrase recovery features should be greater than 1 but less than the ACS quorum as these features can authorize access to OCS/Softcard-protected keys, and this quorum size protects against a single malicious administrator card holder.

5.2.1.1. NSO-timeout guidance

When creating a new Security World, for example using the `new-world` utility, one of the parameters that can be explicitly specified is the NSO-timeout. The NSO-timeout is the maximum time that can elapse between ACS quorum authorization, and reauthorization being required. This is only normally relevant when the last administrator card of the quorum is left in the card reader of a Module, as normal security practice is to remove the last administrator card as soon as the current administrative task has been completed.

The setting of this value will depend on the administrative tasks that the ACS quorum will authorize to be performed in the Security World, and the results of the threat analysis for the nShield administrators/ACS quorum.

The following rules apply to the NSO-timeout:

- The minimum is 3 minutes per card
- The default is the greater of 6 minutes per card, and 1 hour
- The maximum is the greater of 10 minutes per card, and 2 hours

The following is general guidance for NSO-timeout setting. Refer to your own security policy to determine what value is acceptable:

- If the HSM is for general cryptographic use, and key migration may occur from another Security World, the default NSO-timeout is a good choice.
- If the risk of the last administrator card being mistakenly left in the HSM's card reader is considered significant, a shorter value may be set for the NSO-timeout. The value must be set large enough to allow completion of the longest operation the ACS quorum authorize.

5.2.1.2. Creating and maintaining a quorum

Each card set consists of a number of smart cards, N, of which a smaller number, K, is required to authorize an action. The required number K is known as the quorum. Each card in an ACS stores only a share of the ACS keys. You can only re-create these keys if you have access to enough of their shares. Because cards sometimes fail or are lost, the number of shares required to re-create the key (K) should be less than the total number of shares (N).

To make a robustly secure ACS, it is recommended that the value of K is relatively large and the value of N is less than twice that of K (for example, the values for K/N being 3/5 or 5/9). The customer security procedures should determine the values of K and N which should be based on a threat analysis of the protected data.

The customer security procedures should identify the role holders for the different cards. Roles should be assigned based on area of responsibility but also awareness of role holder's availability as they may be required to recover a Security World at short notice in the case of a failure.

The customer's security policy should determine whether the passphrases are required for the cards. passphrases provide an additional barrier to the attacker. This requirement may be necessary based on the value of the data protected and the security around the storage location of cards. A timing delay feature is applied to password retries to add further protec tion.

If an ACS quorum (K of N) is lost, the associated Security World becomes unmanageable as a quorum is needed to replace the old ACS card set. There is no possibility of recovering it.

If a card from an ACS is lost, you should replace the entire ACS as soon as possible.

5.2.1.3. Card management guidance

- ACS cards should be assigned to different stakeholders to prevent a malicious adminis trator attempting to exploit the Security World
- ACS cards should be stored securely in separate locations to prevent an attacker obtaining a quorum of cards
- Passwords should be used to help prevent stolen card(s) being used. To prevent this threat arising do not store passwords with the stored card
- Strong passphrases must be used to prevent an attacker attempting to brute force the password. See Logical token passphrase guidance for guidance on choosing strong passphrases
- A register should be maintained of the administrators and the cards they hold to miti- gate being unable to identify administrators and the cards they hold
- Cardsets should be regularly audited to make sure that they are still available and work- ing. See Audit for further details
- The process for managing loss, theft or corruption of cards should be set out in your security procedures. If a quorum of cards is compromised the Security World pro- tected by the ACS is vulnerable to attack. See Security Incident and Response for fur- ther guidance.
- The requirements for the correct identification, use, movement, storage and protection of cards by trusted, authorized individuals should be set out in your Security Proce- dures to mitigate administrators not adequately protecting the cards they are responsi ble for. See Audit for further details.

5.2.2. Module protection

If module protection is selected then access to any application using the module provides access to module protected keys. Physical and/or logical controls must be applied to the platforms hosting the applications and their environment to prevent unauthorized key access.

5.2.3. Softcard protection

If this option is selected then access to application keys is protected by a softcard logical token and an associated passphrase. The password policy used in accessing the softcards should be stated in the customer's Security Procedures. See Logical token passphrase guid

ance for guidance on choosing strong passphrases.

A register should be maintained of the individuals who have access to softcards to support operation and any role transition.

The customer's security procedures should determine whether a softcard can be recovered, if lost, to a new softcard.

This is enabled by default during Security World creation.

The customer's security procedures should determine whether a softcard passphrase can be replaced if lost. This is disabled by default during Security World creation.

You can use a single softcard to protect multiple keys. A threat analysis will determine the number of keys that should be protected by a softcard.

It is possible to generate multiple softcards with the same passphrase. This option whilst convenient increases the attack surface as an attacker breaking the passphrase will then have access to all keys protected by the softcards. A threat analysis should be performed to determine a safe number of keys that are protected by any softcard and its associated passphrase.

Softcards are persistent; after a softcard is loaded, it remains valid for loading the keys it protects until its KeyID is destroyed. Ensure KeyIDs are destroyed once the required operations are complete.

5.2.4. Logical token passphrase guidance

The following public sources of passphrase guidance are recommended as references for creating a user password policy:

- National Cyber Security Centre (UK) - Password Guidance Summary
- Appendix A of NIST Special Publication 800-63B - Digital Identity Guidelines
- SANS Password Construction Guidelines

A timing delay feature is applied to password retries to add further protection, however there is no retry lockout. Therefore, implementing a robust user password policy helps mitigate a determined passphrase attack. In support of this, a warning message can be configured if passphrases are too short and don't comply with your security procedures. The warning message can only be enabled during Security World creation (but then applies to OCS and softcard passphrase entry too within that world). Note that this is an informational check only.

The process for managing forgotten passphrases should be set out in your security proce-

dures.

The lifecycle for passphrases will be determined by your threat analysis and the resulting Security Policy.

5.2.5. OCS protection

OCSs provide the tightest control over application key usage, especially when they are also passphrase protected. Token protected keys use physical tokens in the form of smart cards (ISO 7816 compliant). These belong to a specific Security World and only an HSM within the Security World to which the OCS belongs can read, erase or format the OCS. There is no limit to the number of OCSs that you can create within a Security World. OCSs can be created and deleted at any time.

5.2.5.1. Creating and maintaining a quorum

Each card set consists of a number of smart cards, N, of which a smaller number, K, is required to authorize an action. The required number K is known as the quorum. Each card in an OCS stores only a fragment of the OCS keys. You can only re-create these keys if you have access to enough of their fragments. Because cards sometimes fail or are lost, the number of fragments required to re-create the key (K) should be less than the total number of fragments (N).

To make a robustly secure OCS, it is recommended that the value of K is relatively large and the value of N is less than twice that of K (for example, the values for K/N being 3/5 or 5/9). The customer security procedures should determine the values of K and N which should be based on a threat analysis of the protected data.

The customer security procedures should identify the role holders for the different cards. Roles should be assigned based on area of responsibility.

The customer's security policy should determine whether the passphrases are required for the cards. passphrases provide an additional barrier to the attacker. This requirement may be necessary based on the value of the data protected and the security around the storage location of cards. A timing delay feature is applied to password retries to add further protec tion.

The customer's security policy should determine whether persistent or non-persistent (default) mode is required and whether a time-out is required. See Persistence and non-per sistence for OCS for guidance on this option. Once set at creation the mode cannot be changed.

The customer's security procedures should determine whether OCS can be recovered if lost to a new OCS. This is enabled by default during Security World recreation.

The customer's security procedures should determine whether OCS passphrases can be replaced if lost. This is disabled by default during Security World recreation.

Lost or damaged cards should be replaced as you discover the loss or damage to prevent a potential scenario where a quorum of cards are not available to authorize operations.

For further guidance on using an OCS to share keys across HSMs and share keys between users, see Creating and maintaining a quorum.

5.2.5.2. Card management guidance

- When not in use OCS cards must be stored securely by each custodian.
- See Logical token passphrase guidance for guidance on choosing strong passphrases.
- passphrases or hints for each specific card should not be written down in the same location as the card.
- A register should be maintained of the custodians and the cards they hold to support operation and any role transition. In pursuit of this OCSs can be created with a name of the OCS and names for each card in the OCS.
- Cardsets should be regularly audited to make sure that they are still present. See Audit for further information.
- Sometimes an OCS may be stored in the card reader. See Persistence and non-persistence for OCS for guidance on this option.
- The process for managing loss, theft or corruption of cards should be set out in your security procedures. If a quorum of cards is compromised the application keys protected by the OCS are vulnerable to attack. See Security Incident and Response for fur ther guidance.
- The requirements for the correct identification, use, movement, storage and protection of cards by trusted, authorized individuals should be set out in your security procedure. See Audit for further details.

5.2.5.3. Persistence and non-persistence for OCS

If you create a standard (non-persistent) OCS, the keys it protects can only be used while the last required card of the quorum remains loaded in the smart card reader of the nShield HSM. The keys protected by this card are removed from the memory of the HSM as soon as the card is removed from the smart card reader. This mode is more secure as the user directly controls key usage. If you want to be able to use the keys after you have removed

the last card, you must make that OCS persistent. Keys protected by a persistent card set can be used for as long as the application that loaded the OCS remains connected to the HSM (unless that application removes the keys explicitly or any usage or time limit is reached). Persistent mode should only be used once a threat analysis of the environment has determined that it is safe for application keys to continue to be operationally usable once the last OCS card has been removed.

OCSs (both persistent and non-persistent) can also be created with a time-out, so that they can only be used for limited time after the OCS is loaded. Keys will be forcibly unloaded when the timeout expires. An OCS is loaded by most applications at start up or when the user supplies the final required passphrase. After an OCS has timed out, it is not loadable by another application unless it is removed and reinserted. The removal and then reinsertion of OCS cards is not enforced for dynamic slots. A TVD or passphrase has to be used ensure user interaction before the token is reloaded.

Time-outs operate independently of OCS persistence.

You can manually remove all keys protected by persistent cards by clearing the HSM. For example, you could:

1. Run the command:

   ```
   nopclearfail --clear --all
   ```

2. Press the **Clear** button of the HSM
3. Turn off power to the HSM.

A persistent OCS with no timeout is suitable for a web server. However, using this option is dependent on the level of security of any running application. For example anyone that is able to gain unauthorized application access can use the key.

A non-persistent OCS with no or a short time-out would be suitable for a root Certificate Authority. It provides complete control over key availability – the key is unloaded when the card is removed from the card reader and becomes inactive after an assigned timeout, if it has been mistakenly left in the card reader.

A threat analysis should determine which configuration of persistence/non-persistence/time-out/no time-out is appropriate for the various sets of keys protected by OCSs.

5.2.5.4. Application independence

Although keys belong to specific client applications performing different functions (with possibly different sensitivities), OCSs do not. You can protect keys for different applica-

tions using the same OCS. However, you must not use the same OCS to protect keys for many different applications as a compromise of the OCS could lead to a compromise of all application keys protected by it. Assigning different OCSs to different applications mitigates this threat. Additionally assigning more than one OCS to an application key helps maintain operation in the event of a compromise against an OCS.

5.3. Application keys

When you generate an nShield key (or create it from imported key material), that key is associated with an HSM-enforced Access Control List (ACL). This ACL prevents the key from being used for operations for which it is unsuited, and can enforce requirements that certain tokens be presented, before the key can be accessed. For example, the ACL can specify that a key can only be used for signing, with a specific signing mechanism/algorithm. Your threat analysis will determine what ACL settings are required for a particular key.

5.3.1. ACL restrictions for key wrapping/encapsulation keys

Care must be taken with setting the ACL for all key wrapping/de-encapsulation keys, that they are assigned the single purpose of key wrapping/de-encapsulation, and not allowed to be used for any other purpose. If a wrapping (or de-encapsulation) key is also assigned the decrypt permission, this could lead to a wrapped/encapsulated key being exposed in plaintext in the client/host platform.

The ACL will allow other conditions to be specified for a wrapping/de-encapsulation key, that would further restrict its use to:

- A specific wrapping/de-encapsulation mechanism.
- A specific application key that can be wrapped/de-encapsulated.
- Specific parameters used for the wrapping/de-encapsulation mechanism.

5.4. nShield Connect front panel

In the case of the nShield Connect, HSM configuration can occur through the front panel. You can control access to the menus on the unit and the Power button on the front panel by using **System > System configuration > Login settings**.

When **UI Lockout with OCS** has been enabled, you must log in with an authorized Operator Card before you can access the menus. You can still view information about the unit on the startup screen. When you are logged in, you can log out and leave the unit locked. An OCS to be used to authorize login on a unit must be persistent and not loadable remotely. In

accordance with the principle of 'separation of duties', the OCS, which protects physical access to the nShield Connect HSM, should **not** be used to also protect application keys.

When **UI Lockout without OCS** has been enabled, you cannot access the menus, but you can still view information about the nShield Connect on the start-up screen.

The power button lockout can be enabled and disabled independently when UI Lockout allows access to the menus.

Customer security procedures should identify the settings for the front panel based on a threat analysis of the environment.

5.5. Configuring remote administration access to nShield Connect

A privileged connection is required to perform certain administrative tasks on the nShield Connect, for example to initialize a Security World or to perform certain remote configuration operations. If privileged connections are allowed, the client can issue commands (such as clearing the HSM) which interfere with its normal operation. We recommend that you add only unprivileged clients for normal operational application usage unless the machines need access to administrative operations (such as clearing module, or remotely configuring CodeSafe 5 in the case of 5c). The RFS machine is a privileged client by default (provided it is authenticated with a KNETI network authentication key), and also has privilege to push updated nShield Connect `config` files (in addition to any config-push client specified in `[config_op]` configuration section). The RFS machine is therefore recommended to be used as the general administration client, and kept separate from normal operational usage for crypto operations, but it is also possible to disable the RFS being a privileged client and con fig-push client in the Connect config file.

5.6. Role holder lifecycle guidance

5.6.1. Roles

The roles that can access the applications that use the HSM and their access rights for using application keys should be identified in your security procedures. Access rights must be assigned as the minimum required for a role to be performed.

5.6.1.1. Windows user privileges

Maintaining the integrity of your system against deliberate or accidental acts can be

enhanced by appropriate use of (Operating System (OS)) user privileges. There are two levels of user distinguished by default in nShield Security World on Windows:

- Built-in Administrators group access, running with elevated privilege (note that the name of this group differs depending on the language of the Windows operating system installation)
- Normal local users.

Built-in Administrators group is required for such tasks as:

- Software installation, starting and stopping the hardserver and SNMP
- Privileged connections to nFast Server (hardserver) service
- nShield 5 administration operations with `hsmadmin` and `csadmin`
- Editing the `config` file

By default, normal users can not create Security Worlds and cannot modify other users keys, though they will do read access to them and can create their own. Encrypted copies of keys are held in Key Management Data (`C:\ProgramData\nCipher\Key Management Data\local`). File permissions can be altered to restrict access to specific keys to specific users/groups, or an alternative Key Management Data (kmdata) folder can be specified by using the `NFAST_KMDATA_LOCAL` environment variable, which would allow the Windows access controls of the directory to be controlled independently by the administrator of the system. Note that the nShield Security World installer sets access controls on installation directories, so changes may not persist when upgraded or re-installed. Both the unprivileged and privileged connections to the nFast Server (hardserver) can have their group access specified in the `nt_pipe_users` and `nt_privpipe_users` fields in the `config` file (note that these settings control access for clients using both domain sockets and local TCP connections). This enables the unprivileged access to be restricted (from all local users) and the privileged access relaxed (from Built-in Administrators running with elevated privilege).

nShield services in Windows run with the minimum privilege that they require. Due to the privileged operations done by the nFast Server (hardserver) service it runs as LocalSystem. Other nShield services (such as nFast Remote Administration Service and nCipher SNMP Agent) run as a virtual service account (automatically managed by the OS) and also drop all unnecessary Windows privileges during startup. The nFast Server (hardserver) service automatically gives the appropriate level of access to known nShield services even if the `nt_pipe_users` and `nt_privpipe_users` fields in the `config` file are overridden.

5.6.1.2. Linux user privileges

Maintaining the integrity of your system against deliberate or accidental acts can be

enhanced by appropriate use of (OS) user privileges. There are four types of user:

- Superuser (`root`)
- `nfastadmin` group user
- `nfast` group user
- Normal users.

Typically, normal users can carry out operations involving Security Worlds, cardsets and keys, but not create Security Worlds, keys and cardsets. `nfast` group users have enhanced access, enabling them to create Security Worlds, cardsets and keys. Encrypted copies of keys are held in `kmdata` (`/opt/nfast/kmdata/local`). Normal users only have read access to the files, whereas `nfast` group users have read and write access, enabling them to create and use keys. `nfast` group users can also make privileged connections to the hardserver, e.g. enabling them to clear or change the mode of an HSM, and perform some administration operations on an nShield Connect (provided that the client they are running on is also a privileged client of the nShield Connect). The access to `kmdata` role and the privileged (local) client role can be separated by using a non-default `kmdata` path by setting the `NFAST_KMLOCAL` environment variable, and setting alternative user or group access to the path that it specifies. `nfastadmin` group users by default are permitted to do certain `hsmadmin` configuration and monitoring operations for the nShield 5, and `csadmin` configuration and monitoring operations for CodeSafe 5. Non-default groups can be specified instead for the different nShield 5 service roles by setting override environment variables at install time.

Superuser access (`root`) is required for such tasks as software installation, and starting and stopping services such as the hardserver and SNMP.

5.6.2. Access rights withdrawn

Customer Security Procedures should identify the requirements and timelines for rescinding access rights. These may be for the following reasons:

- When a role holder leaves the company
- Moves department
- Changes roles
- Is long term sick
- Is suspended from duties.

5.6.3. Dos and don’ts for access control mechanisms

Most failures of security systems are not the result of inherent flaws in the system but result from user error. The following basic rules apply to any security system:

- Keep your smartcards safe.
- Always obtain smartcards from a trusted source: from Entrust or directly from the smartcard manufacturer.



nShield Remote Administration Cards can only be supplied by Entrust.

- Never insert a smartcard used with nShield HSMs into a smartcard reader you do not trust.
- Never connect a smartcard reader you do not trust into your HSM.
- Do not enter passphrases into a server that you do not trust.
- Never tell anyone your passphrase.
- Use a strong passphrase, see Logical token passphrase guidance.
- It is recommended that Remote Administration cards have the optional passphrase implemented to prevent the Logical Token share on the remotely presented card being misused by a malicious or misconfigured client of the HSM.

6. Operation

For additional procedural guidance on operational issues (describing how the HSM should be initially configured/setup), see Commissioning. Refer to that chapter as well for operational guidance.

6.1. Patching policy

A patching policy should be defined and actively implemented. Specifically:

- The latest version of the nShield firmware/software should be installed.
- Any host operating systems should be a current, supported version with the latest patches applied.
- It is recommended that anti-virus software is installed on the host system and maintained with automatic updates.

6.2. Set the RTC time

Set the nShield Edge, nShield Solo and nShield Connect RTC using an accurate trusted local time source at regular intervals to mitigate any clock drift.

6.3. Operator Card Set (OCS) quorum configurations

6.3.1. Share keys across multiple HSMs

An OCS can enable the same keys for use in a number of different HSMs at the same time.

However, it does mean that if you have a non-persistent OCS, you have to leave one of the cards in an appropriate card slot of each HSM. If you have created an OCS in which K is more than half of N, you can share the keys it protects simultaneously amongst multiple modules as long you have enough unused cards to form a K/N quorum for the additional HSMs. For example, with a 3/5 OCS, you can load keys onto 3 HSMs because, after loading the key on the first device, you still have 4 cards left. After loading the key on a second device, you still have 3 cards left. After loading the key onto a third device, you have only 2 cards left, which is not enough to create the quorum required to load the key onto a fourth device.

However, in this instance, the guidance outlined in Access Control regarding security controls for cards is not implemented. The loss or theft of the quorum of cards means that all

keys protected by the OCS are vulnerable. Therefore, a threat analysis should identify the additional logical and physical controls required to protect the OCS left in the card reader.

Alternatively a less secure method of using OCS-protected keys across multiple HSMs is to set:

- K to 1
- N at least equal to the number of the HSMs you want to use.

You can then insert single cards from the OCS into the appropriate card slot of each HSM to authorize the use of that key.

To mitigate the risk of card failure consider setting N to a greater number than the number of HSMs. In the event of failure the spare OCS card is retrieved from its secure location and is deployed whilst arrangements are made to create a new OCS to replace the existing one.

However, the guidance outlined in Creating and maintaining a quorum regarding quorum ratios and security controls for cards is not implemented. The misuse, loss or theft of one card means that all keys protected by the OCS are vulnerable. Therefore, a threat analysis must identify the additional logical and physical access controls required to protect the OCS left in the card reader.

An alternative strategy to the configurations listed above is to use a persistent OCS or a per sistent OCS with a time-out. However, both of these options reduce the control the user has over keys once the OCS has been loaded. A threat analysis should determine which con figuration of persistence/non-persistence/time-out/no time-out is appropriate for the vari- ous sets of keys protected by OCSs.

6.3.2. Share key between users

To share the same OCS-protected key to a set of users, set:

- K to 1
- N equal to the number of users.

You can then give each user a single card from the OCS, enabling those users to authorize the use of that key. However, in this instance, the guidance outlined in Creating and main- taining a quorum regarding quorum ratios for cards is not implemented. The misuse, loss or theft of one card means that all keys protected by the OCS are vulnerable. Therefore, a threat analysis should identify the additional logical and physical controls required to pro- tect the loss of one card.

6.4. NVRAM key storage

Application keys can be stored within the nonvolatile memory of a suitable HSM. This functionality is provided exclusively for regulatory reasons. NVRAM-stored keys provide no additional security benefits and their use exposes your ACS to increased risk. Storing keys in nonvolatile memory also reduces load-balancing and recovery capabilities. Because of these factors, we recommend you always use standard Security World keys unless you are explicitly required to use NVRAM-stored keys.

Any backup and recovery procedures for NVRAM-stored keys must be consistent with regulatory requirements. Do NOT back-up keys to a smart card, as the keys would no longer be stored solely within the physical boundary of the HSM.

6.5. Config push feature

The config push feature allows the nShield Connect configuration to be updated remotely, that is, without access to the nShield Connect front panel. Therefore, anyone with access to the RFS and/or designated client can change the nShield Connect configuration using cfg-pushnethsm. If config push is not required, it should be disabled via the nShield Connect front panel. If this feature is required, conduct a threat analysis to determine if this option is a security risk and confirm if any security controls are required.

6.6. Security World replacement options

If you replace an existing Security World, its `%NFAST_KMDATA%\local` directory is not overwritten but renamed `%NFAST_KMDATA%\local_N` (where *N* is an integer assigned depending on how many Security Worlds have been previously saved during overwrites). A new Key Management Data directory is created for the new Security World. If you do not wish to retain the `%NFAST_KMDATA%\local_N` directory from the old Security World, you must delete it manually.

6.7. Host platform and client applications

The nShield security model assumes that the security of the client endpoint, including any client applications, is completely under the customer's control, and that the host platform is physically protected and hardened in accordance with the customer's Security Policy. Additional security controls may be put in place to reduce the client machine's attack surface, including file encryption and implementation of mandatory access controls. However, these are entirely at the discretion of the customer deploying the system, and will be guided by

their threat analysis.

The following assumptions are made of the host platform and client application to ensure secure operation of the HSM:

- Client applications, users running nShield utilities, and any user on the host platform are assumed to be trusted, and will not perform malicious actions which would be detri mental to the security of the HSM, or the security concerns of the HSM's operator.
- The administrator of the host platform should ensure that only authorised and trusted users are allowed access to the HSM's nCore API, and that access to the HSM's privi- leged nCore API is only provided to users that definitely need access to these HSM security significant commands.
- The administrator of the host platform should implement platform service user separa- tion (as outlined in SSH Client Key Protection (nShield 5s HSMs)), so that only specific authorised and trusted users will have access to nShield 5 platform services.
    - It is recommended that ssh private authentication keys, on the host platform, for the different nShield 5 platform services are encrypted, as outlined in SSH client key encryption.
- The administrator of the host platform should ensure that it has any necessary security patches loaded, is free from unapproved software, and is protected against malware.
- Any client application using the cryptographic services of the HSM should securely gather identification and authentication data from its users and securely transfer it to the HSM when authorizing the use of HSM assets and services.
- Any client application using the cryptographic functions of the HSM should ensure that the correct data are supplied in a secure manner (including any relevant require- ments for authenticity, integrity and confidentiality).
- Any client application using the cryptographic functions of the HSM should protect the security concerns of its users, and comply with the user's security policies concern ing key and sensitive data management.
- The client should ensure that the host platform running their application contains an available random number generator of sufficient quality to allow the generation of keys that contain enough entropy as to be unpredictable.
- The client should ensure that all cryptographic keys generated or cryptographic algo- rithm used on the host platform conform to specifications that are endorsed by recog- nized security authorities.
- The strength of client keys used to build a secure channel to the HSM must meet the strength specified in NIST SP800-131A Revision 2.

6.8. Preload utility

The `preload` utility preloads keys, for example, when using a PKCS #11 application. For a description, see Preload Utility.

The use of `preload` is **not** recommended if any of the following apply on the client/host plat form:

- Standard user or group file permissions, that is, file discretionary access control, can be circumvented. For example, on Linux, via users being permitted to execute `sudo` commands.
- The threat exists that a malicious user (attacker) can gain unauthorized access to the user's account on the client platform. For example, if the remote login authentication credentials are weak for the host platform.
- Malicious insider threat: A privileged user of the client platform is untrusted, and could bypass security controls that may lead to compromising the security of other users on that client platform.

6.9. Discarding keys

To destroy a key permanently you must either erase the OCS that is used to protect it or erase the Security World completely. There are no other ways to destroy a key permanently.

6.10. Erasing a module from a Security World

You do not need the ACS to erase a module. However, unless you have a valid ACS and the host data for this Security World, you cannot restore the Security World after you have erased it.

6.11. Replacing an OCS

Replacing an OCS requires authorization from the ACS of the Security World to which it belongs. You cannot replace an OCS unless you have the required number of cards from the appropriate ACS.

Replacing one OCS with another OCS also transfers the keys protected by the first OCS to the protection of the new OCS.

When you replace an OCS or softcard and recover its keys to a different OCS or softcard, the key material is not changed by the process. The process deletes the original Security

World data (that is, the encrypted version of the key or keys and the smart card or softcard data file) and replaces this data with host data protected by the new OCS or softcard.

Keys protected by an OCS can only be recovered to another OCS, and not to a softcard. Likewise, softcard-protected keys can only be recovered to another softcard, and not to an OCS.

We recommend that after you have replaced an OCS, you then erase the remaining cards so that the old card set's logical token can never be recovered.

Deleting the information about an OCS from the client does not remove the data for keys protected by that card set, as this data may exist in a backup, or the RFS.

If you are sharing the Security World across several client computers, you must ensure that the changes to the host data are propagated to all your computers.

6.12. Replacing the ACS

Replacing the ACS modifies the world file. In order to use the new ACS on other machines in the Security World, you must copy the updated world file to all the machines in the Secu rity World after replacing the ACS. Failure to do so could result in loss of administrative access to the Security World.

We recommend that you erase your old Administrator Cards as soon as you have created the new ACS. An attacker with the old ACS and a copy of the old host data could still re-cre ate all your keys. With a copy of a current backup, they could even access keys that were created after you replaced the ACS.

6.13. Firmware upgrade

Verify the firmware installation media before you load firmware from it. See Software and firmware integrity.

If you are upgrading a module which has SEE program data or NVRAM-stored keys in its nonvolatile memory, use the NVRAM-backup utility to backup your data first.

6.13.1. Setting the minimum VSN after a firmware upgrade

For nShield 5, the `hsmadmin setminvsn` command will set the minimum VSN (minVSN) for the loaded firmware. If an attempt is made to perform a firmware update where the VSN of the updated firmware is less than the minVSN, the firmware update will be refused. The `min`

`VSN` will be set at Module manufacture to the VSN of the Module's initial firmware.

Whenever new authorised firmware has been loaded and tested, it is recommended that the `hasmadmin setminvsm` command is used to set the minVSN to the VSN of the newly loaded firmware. This will protect the HSM against a firmware rollback/downgrade attack. For more information about the nShield 5 `hasmadmin setminvsm` command, see hsmadmin setminvsn.

6.14. Enabling and disabling remote upgrade

You can enable or disable remote upgrade for an nShield Connect. If remote upgrade is not required, this option must be disabled.

6.15. Migrating keys to a v3 Security World



A v3 Security World is a Security World created using a v12.50 (or later) nShield release.

In additions to the guidance provided in the User Manual, it is strongly recommended that the following security related guidance is followed when performing key migration to a v3 Security World:

- Perform the key migration in a controlled environment, where the Source and Destination HSMs are logically and physically isolated from all possible external influences. This will reduce the risk of an external party manipulating the data that is defining the Destination Security World/HSM
- Explicitly initialize the Source and Destination Security Worlds onto the Source and Destination HSMs, so that both the migration tool operator and ACS quorums can have assurance that the correct Security Worlds are being used, and the keys will be migrated to the correct Destination Security World
- The ACS holders (for the Source Security World) should verify the parameters used in the migration tool to ensure those are correct, before allowing key migration to proceed. For example, it is very important that the identifiers for the Destination HSM are entered correctly, so that the keys are migrated to the correct Destination Security World.

If application key are migrated to a v3 Security World, it should be noted that the security strength of any migrated keys shall be equal to the minimum security strength of any of the following:

- The security strength of the application key to be migrated
- The security strength of the Source Security World
- The security strength of the Destination Security World (which for a v3 Security World will be 128bits)

It should also be noted that during the procedure of migrating keys to a v3 Security World, when the migrated keys are outside the nShield HSM's protected boundary, these keys are always encrypted with keys of >112 bits security strength.

6.16. Untrusted input validation

For example, HCM-encrypted data should contain an authentication tag, however if you receive it from an untrusted source, you should verify it against a known trusted value before use. In this situation the length of the data source authentication tag will normally be present in the received data, but this should not be implicitly trusted, but verified against a known trusted value before use. This would mitigate against any manipulation of the received ciphertext, which is attempting to reduce the security strength of the data source authentication protection.

7. Key Management

Key management is a critical component of a security product. A risk analysis will determine the strength of cryptographic algorithm and associated key sizes and lifetimes required to protect customer data.

7.1. Key management schema

Refer to Security World infrastructure for a description of the Security World infrastructure available for managing the secure life-cycle of cryptographic keys. See Security World Access Control Architecture for a description of the different keys available to protect appli cation keys.

7.2. Security World security strengths

Security strength is represented by a number associated with the amount of work (that is, the number of guesses/operations) that is required to break a cryptographic algorithm or system. The security strength is specified in bits and is a specific value from the set {80, 96, 112, 128, 192, 256}. For example, a security strength of 112 bits would require on average $2^{(112-1)}$ operations to brute force the algorithm or system.

Security World modes are selected when creating a Security World. The available modes, associated cipher suites, automatic FIPS mechanism compliance and security strengths are:

Modes	Ciphersuite	Automatic compliance of FIPS mechanisms with NIST SP800-131A Revision 1 algo-rithm/key sizes	Security Strength
FIPS 140 Level 3 (FIPS approved mode enforced)	ECp521mAES DLf3072s256mAESc-SP800131Ar1	Yes	128 bits
Unrestricted (default)	DLf3072s256mAESc-SP800131Ar1	No (see note 1)	128 bits
Common Criteria CMTS (see note 2)	DLf3072s256mAESc-SP800131Ar1	No	128 bits



1. In this mode, non-approved security functions, for example algo-rithms and primes, that are not compliant with NIST SP800-131A Revision 2 are available, however if selected then you will be operat

ing outside of a FIPS approved mode of operation. See the FIPS-140-2/3 specification for a description of a FIPS approved mode of operation).

2. Common Criteria EN 419 221-5 Protection Profiles for TSP Cryptographic Modules - Part 5 Cryptographic Module for Trust Services.

Security Worlds created with a pre v12.50 release can be loaded. The cipher suites, automatic FIPS mechanism compliance and security strengths for these Worlds are:

Pre v12.50 Security Worlds ciphersuites, FIPS mechanism conformance and security strengths:

Modes	Ciphersuite	Automatic compliance of FIPS mechanisms with NIST SP800-131A Revision 1 algorithm/key sizes	Security Strength
Not Applicable	DLf3072s256mRijndael	No	128 bits (see note 3)
	DLf1024s160mRijndael	No	80 bits
	DLf1024s160mDES3	No	80 bits

3. Whilst the Ciphersuite provides 128 bits of security strength, some of the underlying (not selectable) cryptographic mechanisms use by this Ciphersuite are no longer FIPS approved. This is identified by the term "No" in the "Automatic compliance with FIPS mechanisms" column.

The tables above identify the ciphersuites, automatic compliance with NIST SP800-131A Revision 1 and security strength. The National Institute for Standards and Technology (NIST) have published, over the years, good key management guidance. NIST SP800-131A Revision 1 – *Transitioning the Use of Cryptographic Algorithms and Key Lengths* is referenced in the table as it provides guidance on suitable cryptographic algorithms and key sizes for protecting sensitive data today. This document, published in November 2015, advises that the minimum security strength for algorithms or keys is now 112 bits. Whilst the immediate audience is U.S. government agencies, NIST standards provide a global benchmark in security standards which many global product vendors adhere to, in order to provide their customers with appropriate levels of security assurance. Therefore, the industry standard minimum security strength is considered to be 112 bits today.

7.2.1. KML type and security strength

The Security World security strength may be impacted by selecting a KML type which is

not the default for the cipher suite (see KML type selection). The motivation for this is to improve performances, but with a potential security tradeoff.

There are currently two valid scenarios:

- Selecting the ECDSA `NISTp256hSHA1` KML type with a DSA-3072 Security World:

  There may be a performance boost from using NIST P-256 with no security tradeoff.

  This configuration is not supported in FIPS 140 Level 3 mode.

- Downgrading the KML type from `NISTp521hSHA1` to `NISTp256hSHA1` with an ECDSA Security World:

  There is a performance boost from using NIST P-256 instead of P-521 but with a security tradeoff. This downgrades the KML security from 256-bit security to 128-bit security.

  This configuration is approved in FIPS 140 Level 3 mode.

7.3. Application keys algorithms and key sizes

Depending on the application library used, a range of cryptographic algorithms are available for selection. The algorithm used and key size selected (if applicable) should be sufficient to protect customer data from threats identified in the deployed environment for their data. In line with standard security best practice, the security strength, as described in Security World security strengths, of the Security World ciphersuite will effectively limit the security strength that can be claimed for any key or algorithm used by the HSM. As advised in Security World security strengths a minimum security strength of 112 bits is considered the industry standard.

Cryptographic algorithms identifies all algorithms and key sizes available (both NIST approved and non approved). NIST SP 800-57 Part 1 Revision 4 has a section on Comparable Algorithm Strengths which provides guidance on identifying the security strength of different NIST approved algorithms and key sizes. BlueKrypt Cryptographic Key Length Recommendation could be a useful reference for determining required key sizes for common cryptographic functions:

- Symmetric algorithms
- Asymmetric algorithms, based on the following branches of cryptography:
  - Integer Factorization as a branch of Integer Factorization Cryptography as in NIST SP800-56B, such as RSA.
  - Discrete Logarithm as the branch of Discrete Logarithm Cryptography (see NIST

SP800-56A), such as DSA, Diffie-Hellman.

- Elliptic Curve, such as ECDSA.

- Hashes.

The *General Key Management Guidance* section of NIST SP 800-57 Part 1 Revision 4 provides guidance on the risk factors that should be considered when assessing cryptoperiods and the selection of algorithms and keysize.

The `nfkmverify` command-line utility can be used to identify algorithms and key sizes (in bits). See Cryptographic algorithms for more information.

7.4. Cryptoperiods

A cryptoperiod is the time span during which a specific cryptographic key is authorized for use.

NIST SP 800-57 Part 1 Revision 4 has sections describing the rationale for cryptoperiods and the risk factors that should be considered when determining cryptoperiods. Setting a cryptoperiod limits, amongst other things, the amount of data exposure if a key is compromised. You are advised to perform a threat analysis, considering the sensitivity of your data and what controls you have in place to mitigate compromise, to determine appropriate cryptoperiods for keys protecting that data.

In terms of the key management schema the following cryptoperiod rules must be applied:

- Application keys must have a cryptoperiod assigned to them
- The Security World must have a cryptoperiod, as the Security World keys (for example, module keys) are used to protect application keys. The cryptoperiod of the Security World must therefore be greater than or equal to the cryptoperiod of any application key.
- Whenever an application key reaches the end of its cryptoperiod it must be revoked. Whenever a Security World reaches the end of its cryptoperiod, a new Security World must be initialized under a new ACS and new application keys generated. The old Security World's ACS must be destroyed.
- Security World application keys can be created with a timeout and/or a usage value to help manage cryptoperiods. Additionally in Common Criteria CMTS mode application keys can use max timeout and max usage parameters to manage cryptoperiods. Note that softcard protected application keys can't set a usage or timeout limit as there is no option to set this in the ppmk utility. The cryptoperiods for infrastructure keys can be managed manually by deleting a key when it has reached the end of its cryptoperiod. This may require re-initialization of the Security World.

7.5. Generating random numbers and keys

The nShield HSM includes a certified random number generator that uses a hardware based source of entropy. This provides greater security than a random number generator that uses a non-hardware based source of entropy that is typically provided by general purpose computers. Therefore, always use the nShield HSM to generate random numbers and keys.

7.6. Key backup

The sensitive Security World data, such as application key blobs, stored on the host or RFS is encrypted using the Security World key. You must regularly back up all the data stored in the Key Management Data directory with your normal backup procedures. It would not mat ter if an attacker obtained this data because all sensitive data is protected by the Security World key, stored in your HSM, and the Administrator cards for that Security World.

In terms of cryptoperiods (see above) keys that have reached the end of the cryptoperiod and therefore no longer exist on the nShield HSM may still exist on backups. If feasible then the backup data should also be deleted. However, if the backups have to be maintained for operational, resilience, or audit reasons, then ensure that the relevant procedural controls are implemented to mitigate attacks on retired keys.

7.7. Key import

We recommend generating a new key instead of importing an existing key whenever possible. The import operation does not delete any copies of the imported key material from the host, and as traces of this key material may still exist on disks or in backups, there is a risk that the key material may become compromised after it has been imported. It is your responsibility to ensure any unprotected key material is deleted. If a key was compromised before importation, then importing it does not make it secure again.

7.8. Key separation

7.8.1. Single purpose keys

Key separation, that is, when each key only has a single purpose, is an important security principle which is re-enforced in nShield by having different key types for different purposes, for example, `ECDHPrivate/ECDHPublic` for Elliptic Curve (EC) Key-Establishment keys and `ECDSAPrivate/ECDSAPublic` for EC signing/verification keys).

There is a use case where a static EC private Key-Establishment key will also be used to sign a CSR to request an (initial) certificate for the associated static EC public Key-Establishment key. For this particular use case, the keyType `ECPrivate/ECPublic` should be used for the EC Key-Establishment key, with a specialized ACL allowing the ECPrivate key to be used for a single signing (`OpPermissions:Sign`), and then key-establishment (`OpPermissions:Decrypt`).



When the ACL is used to enforce a single signature operation, this signature must be performed before the key is initially persisted/blobbed.

7.8.2. SSH secure channel keys

Access to each of the platform services is by a separate, mutually authenticated SSH secure channel. Entrust recommends that you use separate key pairs for each service to maintain the key separation principle. This reduces the impact if keys are compromised.

7.9. nShield JCA/JCE CSP

7.9.1. Installing the nShield JCA/JCE CSP

Security configuration guidance for using unlimited strength JCE jurisdiction policy files and the correct preference order for nShield in the Java security configuration file is provided in-situ in the nCipherKM JCA JCE CSP v13.9.9 API Guide. See Installing the nCipherKM JCA/JCE CSP for details.

7.10. nShield PKCS #11 library

7.10.1. Symmetric encryption

The nShield PKCS #11 library can use the nShield HSM to perform symmetric encryption with the following algorithms:

- DES
- Triple DES
- AES.

Because of limitations on throughput, these operations can be slower on the nShield HSM than on the host computer. However, although the nShield HSM may be slower than the host under a light load, you may find that under a heavy load the advantage gained from

off-loading the symmetric cryptography (which frees the host CPU for other tasks) means that you achieve better overall performance.

Performing symmetric encryption on the host increases the threat of key compromise as the security protection provided by the host will be less than the nShield HSM. Additionally there may be a lack of key lifecycle management of the application keys on the host.

For these reasons we recommend performing symmetric operations on the nShield HSM. If symmetric encryption is performed on the host, technical and procedural access controls should be deployed to protect the host, in order to mitigate the higher threat of key compromise.

7.10.2. PKCS #11 library with Security Assurance Mechanism

It is possible for an application to use the PKCS #11 API in ways that do not necessarily provide the expected security benefits, or which might introduce additional weaknesses.

The PKCS #11 library with the Security Assurance Mechanism (SAM), `libcknfast`, can help users to identify potential weaknesses, and help developers create secure PKCS #11 applications more easily.

The SAM in the PKCS #11 library is intended to detect operations that reveal questionable behaviour by the application. This includes applications that use an inadequate concept of key security.

Security guidance on using SAM to detect insecure behavior is provided in-situ in the PKCS #11 Reference Guide for nShield Security World v13.9.9. See PKCS#11 Developer libraries for details.

7.10.3. nShield PKCS #11 library environment variables

Security configuration guidance for various variables is provided in-situ in the PKCS #11 Reference Guide for nShield Security World v13.9.9. See nShield PKCS #11 library environment variables for details.

8. Physical Security

This chapter provides guidance on the physical security controls available on the different nShield platforms and the procedural controls required to maintain those physical security controls across the product's lifecycle.

The industry terms "tamper resistance", "tamper evidence", "tamper detection", and "tamper response", as defined in the glossary, will be used when discussing the physical security con trols available.

8.1. nShield Edge physical security controls

The nShield Edge uses Tamper Resistance and Tamper Evident physical security controls to protect sensitive security parameters within the unit:

The mini HSM within the unit is covered in an epoxy encapsulant to resist tamper attempts. This is a tamper resistant control.



This would also be a tamper evident control as well if it could be inspected. However, the internal mini HSM is additionally protected by the security seal control on the enclosure boundary and therefore can't be inspected.

Once assembled the nShield Edge can't be disassembled without breaking the security seal. This is a tamper evident control.

Once assembled the nShield Edge is difficult to disassemble without disfiguring the fascia. This is a tamper evident control.

See Tamper inspection for procedural control guidance required to maintain tamper evident security controls.

8.2. nShield Solo+ physical security controls

The nShield Solo+ uses Tamper Resistance and Tamper Evident physical security controls to protect sensitive security parameters within the unit:

The nShield Solo+ card is covered in an epoxy encapsulant to resist and provide evidence of tamper attempts.

See Tamper inspection for procedural control guidance required to maintain and manage tamper evident security controls.

8.3. nShield Solo SoloXC and nShield 5s physical security controls

The nShield Solo XC and nShield 5s use Tamper Resistance, Tamper Evident, Tamper Detection and Response physical security controls to protect sensitive security parameters within the unit.

On the nShield Solo XC, a cosmetic metal lid covers the encapsulant and can be removed for inspection purposes. See Tamper inspection for procedural control guidance required to maintain and manage tamper evident security controls.

The nShield Solo XC and nShield 5s feature tamper detection and response mechanisms that indicates the following tamper events:

- Abnormal temperature
- Abnormal voltage
- Low battery voltage (Solo XC only)
- Sensor failure.

When one of the above tamper events is detected, that is, when a possible attempt to compromise the system has been detected, the HSM will perform the following actions:

1. All non-protected secrets in the HSM will be erased
2. The HSM will enter its Error state
3. The particular tamper event is identified by flashing an encoded error on the LED indicator.

After a tamper event, the HSM and its environment should be examined for signs of potential tamper/intrusion, and the tamper event recorded (in accordance with the Customer's Security Incident and Response Policy). If the source of the tamper event can be discovered and can be considered harmless, provided the Customer's Security Incident and Response Policy allows, the HSM can be restarted to bring it back into operation. If the tamper event has not been neutralized, the HSM will just reassert the tamper event.

If the source of the tamper cannot be discovered, then the HSM should be considered to be in a compromised state and will have to either be destroyed or returned to Entrust for secure destruction. See Decommission and Disposal for further information.

For nShield Solo XC hard tamper events and their respective encoded error messages, see Morse code error messages.



In the case of the battery removal/failure tamper event, the tamper event can only be actioned upon the next application of mains power

> (with the battery removed). Once this tamper event is complete (and all non-protected secrets are erased) the HSM should blink the encoded pattern on the LED for low battery voltage <2.5V to indicate that there are no secrets within the HSM to the customer. However, in the unlikely scenario that the HSM is non-functional to the point where the Security Processor will not wake up after mains power is applied, the non-protected secrets will never be erased. Therefore, the customer must monitor for the correct encoded sequence to indicate that all non-protected secrets have been erased. If the LED does not blink the prescribed pattern then the nShield Solo XC must be physically, securely destroyed.

For guidance on how to respond to a tamper, see Security Incident and Response.

8.4. nShield Connect physical security controls

This section provides an overview of the physical security measures that have been implemented to protect your nShield Connect.

You are also shown how to check the physical security of your nShield Connect.

The tamper detection and response functionality on the nShield Connect provides additional physical security, over and above that provided by the tamper evident holographic security seal, and alerts you to tampering in an operational environment. There is a removable lid on top of the nShield Connect, protected by the security seal and tamper detection switches. To prevent the insertion of objects into the nShield Connect, tamper resistant baffles are placed behind vents.

To optimize their effectiveness, use the physical security measures implemented on the nShield Connect in association with your security policies and procedures.

> The FIPS 140 Level 3 cryptographic boundary is at the nShield Solo.

> For more information about FIPS 140, see http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf and https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf.

8.4.1. Tamper event

The nShield Connect offers several layers of tamper protection. The outer boundary (case) of the nShield Connect has tamper detection and response capabilities. When tampered, the unit ceases to provide cryptographic functionality, the operator is informed of the

event, and the unit resets to factory defaults. For guidance on how to respond to a tamper, see Security Incident and Response.

Movements/vibrations, or replacing the fan tray module or a PSU, does not activate the tam per detection functionality.

8.4.1.1. nShield Connect lid is closed

If the nShield Connect is powered, a tamper event has occurred, and the lid is closed, the unit will automatically reset to a factory state. Should this happen, examine your unit for physical signs of tampering, see Physical security checks.

If you discover signs of tampering do not attempt to put the unit back into operation. The date and time of the tamper event are recorded in the log (see Logging, debugging, and diagnostics).



The tamper-responsiveness circuitry has a RTC that runs independently from the main nShield Connect clock. The times associated with events in the tamper log may have slight offsets to times recorded in other log files.

If there are signs of tampering, and the tamper event occurred

- During transit from Entrust, contact Entrust nShield Support
- After installation, refer to your security policies and procedures.

After the unit has reset to a factory state, you require a quorum of the ACS to restore the key data and reconnect the nShield Connect to the network.

8.4.1.2. nShield Connect lid is open

If the nShield Connect is powered, a tamper event has occurred, and the lid is open, the following message is displayed on screen:

```
Unit lid is open
```

An open lid indicates that the physical security of the unit is compromised. You may want to examine your unit for other physical signs of tampering - see Physical security checks. Do not attempt to put the unit back into operation.

The date and time of the tamper event are recorded in the log files (see Logging, debugging, and diagnostics). If the tamper event occurred

- During transit from Entrust, contact Entrust nShield Support
- After installation, refer to your security policies and procedures.

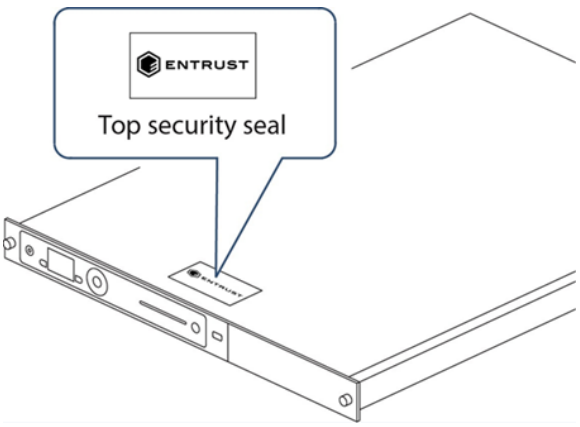
After closing the lid you must reboot the Connect. The unit will then automatically reset to a factory state. If the lid remains open, the above message will remain on the screen, and all button presses are ignored.

8.4.1.3. Physical security checks

An alternative presentation of the physical security checks described here can be found in the *Physical Security Checklist*. For more information about tamper events, and what actions to take if you discover signs of tampering, see Tamper event.

To determine if the security of the nShield Connect is compromised:

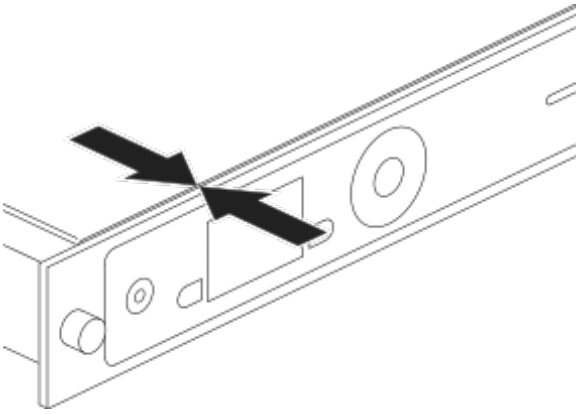
1. Check that the physical security seal is authentic and intact. Look for the holographic foil bearing the holographic tamper label. Look for cuts, tears and voiding of the seal. The seal is located on the top of the nShield Connect chassis.



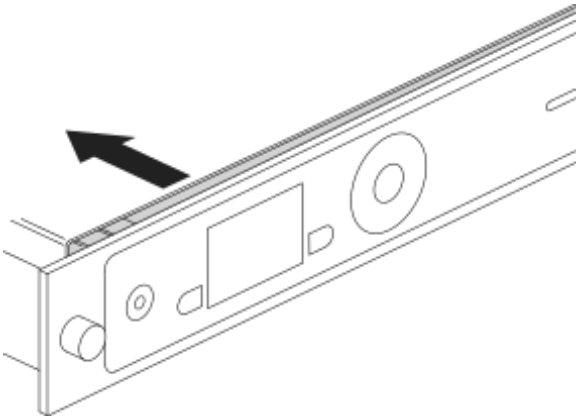
   For information about the appearance of intact and damaged security seals, see the *Physical Security Checklist*.

2. Check that the metal lid remains flush with the nShield Connect chassis.

   Metal lid in the correct position:



Metal lid in an incorrect position (pulled back):

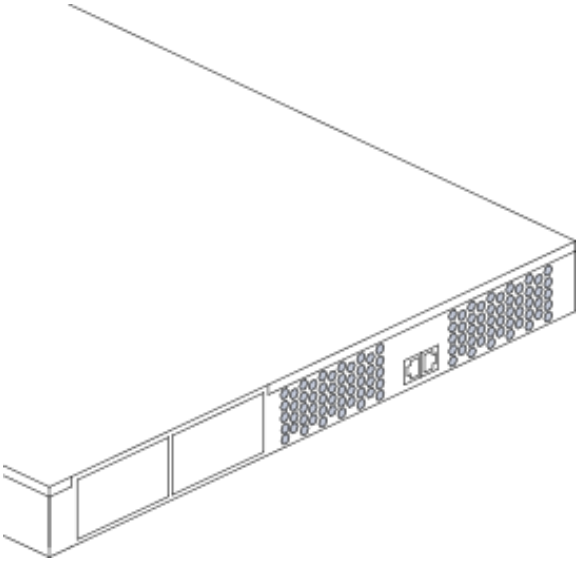


3. Check all surfaces — the top, bottom and sides of the nShield Connect — for signs of physical damage.
4. Check that there are no signs of physical damage to the vents, including attempts to insert objects into the vents.

   Front vent (to the right of the front panel controls):



   Rear vent (to the right of the power supplies):



8.5. nShield card readers

The card reader and its cable should be inspected for signs of tamper.

See Tamper inspection for procedural control guidance required to maintain and manage tamper evident security controls.

8.6. Tamper inspection

Inspections as described in Supply and Transportation. This may include removing the HSM from cabinets to ensure that all surfaces are intact. Further guidance on inspection frequency can be found in Physical inspection. Guidance on how to respond to a tamper can be found in Security Incident and Response.

9. Audit

The product's environment should be audited regularly to ensure that the appropriate set of procedures, satisfying the requirements laid down in this document and any customer Security Procedures, is in place and is being used. A mechanism should be in place to enable corrective action to be taken if any procedure is not being observed or is failing. The Auditor should be independent of the Administrator of the product.

9.1. HSM and card reader location

Customer Security Procedures should state that a record is kept of the location of each HSM and card reader referenced by unique identifiers. This may include its model, serial and any local asset id numbers. This record should be updated if the HSM or card reader is moved.

- Customer Security Procedures should state the frequency for verifying the recorded location of each HSM and card reader.

9.1.1. Physical inspection

Whilst checking the HSM and card reader location, inspections should also be carried to ensure the integrity of the HSM and card reader including any tamper mechanisms as described in Tamper inspection.

- Customer Security Procedures should state the frequency for verifying the integrity of the HSM and card reader including any tamper mechanisms.

9.2. ACS and OCS

Customer Security Procedures should state that a record is kept of either the location (such as in a safe) of each card in an ACS and OCS or the owner depending on the policy stated in the customer's security policy. This record should be updated if a card is moved or transferred.

- Customer Security Procedures should state the frequency for verifying the recorded location or owner each card in an ACS and OCS.

Guidance on how to respond to a missing ACS and OCS cards can be found in Security Incident and Response.

9.3. Logs

Logging and debugging identifies the types of log available across the different nShield plat forms. Each log fulfills a different purpose and some can be filtered to control the amount of information logged. In some instances tamper or cryptographic mechanisms are used to protect the integrity of the logs. Logs that don't use these mechanisms should be protected through procedural controls.

A threat analysis will determine which logs are required and which filters to apply (if available) in monitoring the customer's specific deployment of the HSM.

The Auditor should be independent of the Administrator of the HSM:

- When modifications are made to the configuration of an HSM, the changes should be audited to ensure that the configuration has been modified in the intended way.
- The Auditor should regularly inspect the logs to verify that the unit's configuration reflects the Security Policy.
- The logs should be inspected by the Auditor periodically at a frequency determined by the customer Security Procedures.
- The customer Security Procedures should state what log entries are cause for concern.

The following example scenarios may also be a cause for concern:

- Access outside of work hours
- Unusual changes to the configuration
- Unit power cycled.

The actions required to resolve the issue should also be stated using the customer's own incident response process.

The customer Security Procedures should identify a backup policy for the logs and the authorization required to delete logs once they've been backed-up.

9.4. HSM Audit logging

Audit logging as described in Logging and debugging delivers HSM logs to an external log collector outside of the HSM. It is recommended that audit logging be enabled when creating a Security World. Audit logging uses an integrity mechanism to protect the logs from tampering and verify they generated by a legitimate nShield HSM (as proven by the HSM's KLF2 key and associated KLF2 warrant which certifies its legitimacy). By default, audit logging logs important startup, setup and credential presentation events, but not all key usage, and so is a relatively lightweight option. Individual keys may additionally be specified to be

logged, in which case all usage of them will result in an audit event; this is a higher-volume logging use case and is recommended for high-value low-usage keys such as Root CA keys. Common Criteria Security Worlds have additional audit logging enabled by default compared to ordinary Security Worlds. Additional controls required to support the Audit Log are described in Audit Log. As well as applying the guidance described above, further guidance specific to Audit Logging is supplied here:

The Auditor should inspect the logs to:

- Identify missing logs
- Verify the integrity of logs up to the trusted root
- Identify log entries that are a cause for concern.

Log verification is done with the `cef-audit-verify` tool for the CEF audit logs produced prior to v13.5 firmware, and with the nShield Audit Log Service and `nshieldaudit` client tool for logs produced by v13.5 firmware and later.

9.5. Audit Logging time

The Date and Time provides guidance on correctly setting the time for the nShield Connect clock and RTC. If NTP is enabled then the nShield Connect clock will synchronize to that. Both clock times will appear in the Audit Log and in other logs listed in Logging and debugging. The nShield Connect's clock (if not synchronized to NTP) and RTC are subject to drift and should be regularly set using an accurate trusted local time source.

With regard to the Audit Log we recommend only accepting that the nShield Connect and RTC time stamps represent the actual time if they match each other within a reasonable margin of error.

9.6. nShield 5 Signed System Logs

Separate from the nCore audit logs associated with the Security World, nShield 5 also contains signed system logs associated with the rest of the device operation, such as device startup events and connection attempts and operations with its non-nCore services. This logging is always enabled in nShield 5 since v13.5 firmware and later, and logs are fetched and verified by the nShield Audit Log Service and can be queried using the `nshieldaudit` tool, alongside the nCore logs that these tools also manage.

10. Support and Maintenance

Customers are encouraged to obtain a support contract and regularly update their HSM to the latest firmware release.



Entrust strongly recommend familiarizing yourself with the information provided in the release notes before using any hardware and software related to your product.

10.1. Security advisories

If Entrust becomes aware of a security issue affecting nShield HSMs, Entrust will publish a security advisory to customers. The security advisory will describe the issue and provide rec ommended actions. In some circumstances the advisory may recommend you upgrade the nShield firmware and or image file. In this situation you will need to re-present a quorum of administrator smart cards to the HSM to reload a Security World. As such, deployment and maintenance of your HSMs should consider the procedures and actions required to upgrade devices in the field.



The Remote Administration feature supports remote firmware upgrade of nShield HSMs, and remote ACS card presentation.

Entrust recommends that you monitor the Announcements & Security Notices section on Entrust nShield, https://trustedcare.entrust.com/, where any announcement of nShield Security Advisories will be made.

10.2. Application and Operating System patching

To maintain protection against threats that occur in the system environment operating sys- tems and applications should be updated in accordance with a patching policy as described in Patching Policy.

10.3. Connect fan tray module and PSU maintenance

The nShield Connect contains only two user-replaceable parts:

- The PSUs
- The fan tray module.

Replacing a PSU or fan tray module does not affect FIPS 140 validations for the nShield

Connect, or result in a tamper event. However, in the very rare event that a PSU or fan tray module requires replacement, contact Support before carrying out the replacement procedure.



Do not remove the fan tray for more than 30 minutes, otherwise a tamper event will occur.

For more information about replacing either a PSU or the fan tray module, see the Installation Sheet that accompanies the replacement part or Physical security of the HSM.



Breaking the security seal or dismantling the nShield Connect voids your warranty cover, and any existing maintenance and support agreements.



Mains power plugs on UK cordsets contain a 5A fuse (BS1362). Only replace with the same type and rating of fuse. If a replacement fuse fails immediately, contact Support. Do not replace with a higher value fuse.

If the product has to be moved for maintenance then all movement should occur in accordance with the HSM and Card Reader Location. Similarly, if the nShield Connect is stored before or after maintenance, then, it should be stored in accordance with the guidance outlined in Environment.

10.4. Solo XC fan and battery maintenance

The fan and battery can be replaced should either malfunction or the battery has reached the end of its useful life. Replacing the battery and/or fan whilst the card is powered down will not cause a tamper of any sort. See Physical Security for guidance on tamper events. The replacement procedure is described in Battery replacement and Replace the fan (Solo XC).

If the product has to be moved for maintenance then all movement should occur in accordance with the HSM and Card Reader Location. Similarly, if the Solo XC is stored before or after maintenance, then, it should be stored in accordance with the guidance outlined in Environment.

10.5. Maintenance mode

Firmware upgrades of the module are only allowed when the module is in maintenance mode. Refer to the nShield HSM and Security World product documentation for more information on setting the module into maintenance mode.

10.6. Troubleshooting

In the event of problems with the nShield HSM, refer to the following pages:

- The *Troubleshooting* chapter for your HSM in the HSM User Guide.
- Logging, debugging, and diagnostics.
- The *Status indicators* chapter for your HSM in the HSM User Guide (for PCIe HSMs).

If the problem cannot be resolved contact support.

10.6.1. Debugging information for Java



Debug output contains all commands and replies sent to the hardserver in their entirety, including all plain texts and the corresponding cipher texts as applicable.

10.6.2. Contacting Entrust nShield Support

To obtain support for your product, contact Entrust nShield Support, https://trustedcare.entrust.com/.

11. Security Incident and Response

11.1. Security incident monitoring

The following suspected or actual events or activities should be monitored for:

- Triggering of tamper evident or response functions in the HSM
- Physical non availability of HSM, card reader, card sets, client application, `%NFAST_KMDATA%` folder contents, nShield Connect config file, SIEM collector data, backup data
- Logical non availability of HSM, card reader, card sets, client application, `%NFAST_KMDATA%` folder contents, nShield Connect config file, SIEM collector data, backup data
- Gaps or unexplained entries in the logs, or suspected log tamper
- Evidence of access control violation contrary to any security policy, for example lost token and subsequent logon.
- Evidence of unauthorized use
- Evidence of network attacks on the HSM
- Evidence of excessive performance demands
- Evidence of violation of environmental controls
- Unauthorized changes to configuration settings for HSM and client application, such as updating the module's clock.
- Non-compliance with security process, such as commissioning on an open network
- Non-compliance with security policy, such as using incorrect algorithm strength or con tinuing to use a key outside of its cryptoperiod.

11.2. Security incident management

If a security incident is suspected the Company Security Officer should be alerted immediately and determine which actions must be implemented as advised by your Security Incident and Response Policy. This should cover the following areas:

- Quarantine area, isolate unit and evidence preservation – witnessed snapshot of unit (this should cover determining whether to power off the unit which may result in the loss of evidence against the need to isolate any potential malware resident on the unit)
- Investigation
- Reporting structure and timescales.

11.3. Security incident impact and response

The sections below identifies the impact of various compromises on keys or secrets and the recovery action required.

Under **Recovery Action** the term revoke is used to indicate that the compromised key must no longer be trusted or used. The terms revoke or revocation are normally used in regard to digital certificates (normally containing public keys), where methods exist to indicate that a certified key can no longer be trusted. However, this manual will apply the term to all compromised keys.

11.3.1. Compromised Key or Secret: A brute force attack on blobbed key outside of module

Impact

Application key is compromised and must not be used. The application key will be one of the following types:

- OCS protected application keys
- Softcard protected application keys
- Module/Module Pool protected application keys

Recovery Action

Revoke application key and destroy the Security World, since all application keys in this Security World must now be considered as compromised.

Destruction of the Security World is achieved by erasing/destroying the ACS and re-initializing all the HSMs to a different Security World (with a new ACS).

Alternatively, to mitigate the present threat, the HSMs can be put into pre-initialization mode whilst business recovery procedures are implemented prior to creating a new Security World.

Note that erasing the ACS will prevent a lost/stolen backup being reloaded on to a new HSM.

11.3.2. Compromised Key or Secret: Attacker has subverted memory of HSM

Impact

Application key is compromised and must not be used. The application key will be one of the following types:

- OCS protected application keys
- Softcard protected application keys
- Module/Module Pool protected application keys

Recovery Action

Revoke application key and destroy the Security World, since all application keys in this Security World must now be considered as compromised.

Destruction of the Security World is achieved by erasing/destroying the ACS.

Destroy the HSM as its integrity can no longer be guaranteed.

Create a different Security World on new/different HSMs (with a new ACS).

Note that erasing the ACS will prevent a lost/stolen backup being reloaded on to a new HSM.

11.3.3. Compromised Key or Secret: passphrase for softcard is compromised

Compromise Type

Lost or observed

Impact

The application keys protected by the softcard should be considered as under the control of the attacker

Recovery Action

Revoke application key protected by softcard. If unable to revoke the key, isolate the HSM so that no system can use it.

Erase all copies of blobs associated with the application key protected by softcard in `kmdata/rfs/backups` to prevent attacker trying to use keys with stolen passphrase.

Create replacement application keys under new softcards.

11.3.4. Compromised Key or Secret: A quorum of OCS cards is compromised

Compromise Type

Lost or stolen

Impact

The application keys protected by the OCS are under the control of the attacker

Recovery Action

Revoke application keys protected by OCS.

If unable to revoke keys isolate HSM so that no system can use it. Erase blobs associated with the application keys protected by the OCS in `kmdata/rfs/backups` to prevent attacker trying to use keys with stolen OCS.

Create replacement application keys under new OCS.

If the OCS cards are subsequently recovered they should either be erased or destroyed.

11.3.5. Compromised Key or Secret: A quorum of ACS cards is compromised

Compromise Type

Lost or stolen

Impact

The Security World protected by the ACS should be considered as under the control of the attacker

Recovery Action

Revoke all Module-protected/module pool-protected application keys and recoverable application keys, as these are available to an attacker with a KMData archive, an HSM and the compromised ACS.

Revoke all non-recoverable OCS or Softcard protected keys because they cannot be used without the Security World, which has been compromised and should be destroyed.

Alternatively, to mitigate the present threat, the HSM can be put into pre-initialization mode whilst business recovery procedures are implemented prior to creating a new Security World.

The original Security World must not be used. Destroy the world file and any backups to prevent an attacker recovering the Security World with the ACS, world file and backups.

If the ACS cards are subsequently recovered they should be either erased or destroyed.

Create replacement application keys.

11.3.6. Compromised Key or Secret: Soft KNETI

Compromise Type

Attacker has subverted client memory

Impact

KNETI is compromised and must not be used

Recovery Action

For every Connect that the affected client has communicated with, use the Front Panel or serial console to remove the client's configuration data.

For any RFS that the affected client has communicated with, update the RFS's configuration file to remove the client's configuration data.

Manually delete the kneti file identified as `kneti-hardserver`.

- On Windows, it is stored in `C:\ProgramData\nCipher\hardserver.d\`.
- On Linux, it is stored in `/opt/nfast/hardserver.d/`.

Reboot the client.

Isolate client and investigate compromise.

Once resolved re-configure the Connects/RFS that this client communicated with using same client's IP address/ESN, but the new KNETI hash, and then re-establish the secure channel to the Connect(s)/RFS.

11.3.7. Compromised Key or Secret: nToken KNETI

Compromise Type

A brute force attack on KNETI file held in the Connect's KMData folder.

Impact

KNETI is compromised and must not be used

Recovery Action

For every Connect that the affected client has communicated with, use the Front Panel or serial console to remove the client's configuration data

For any RFS that the affected client has communicated with, update the RFS's configuration filer to remove the client's configuration data.

Manually delete the kneti file identified as `kneti-nToken ESN`.

- On Windows, it is stored in: `C:\ProgramData\nCipher\hardserver.d\`.
- On Linux, is stored in `/opt/nfast/hardserver.d/`.

Destroy the nToken as its integrity can no longer be guaranteed.

If you have an alternative nToken available, configure it to communicate with an nShield Connect. If you do not have an alternative nToken available, you will need to use software-based authentication instead. See Basic HSM, RFS and client configuration for more information.

11.3.8. Compromised Key or Secret: nShield Connect KNETI

Compromise Type

A brute force attack on KNETI file held in the nShield Connect's KMData folder.

Impact

KNETI is compromised and must not be used

Recovery Action

Remove the compromised Connect's data (IP address/ KNETI and H(KNETI)) from any client hardserver's configuration file that has communicated with the compromised Connect.

Destroy the nShield Connect as its integrity can no longer be guaranteed.

Configure a new nShield Connect to communicate with a client.

11.3.9. Compromised Key or Secret: Soft KNETI

Compromise Type

A brute force attack on KNETI file held in obfuscated form in the KMData folder

Impact

KNETI is compromised and must not be used

Recovery Action

For every Connect that the affected client has communicated with, use the Front Panel or serial console to remove the client's configuration data.

For any RFS that the affected client has communicated with, update the RFS's configuration filer to remove the client's configuration data.

Manually delete the kneti file identified as `kneti-hardserver`.

- On Windows, is stored in `C:\ProgramData\nCipher\hardserver.d\`.
- On Linux, is stored in `/opt/nfast/hardserver.d/`.

Reboot the client.

Isolate client and investigate unauthorized access to the KMData file and the integrity of the client.

Once resolved re-configure the Connects/RFS that this client communicated with using same client's IP address/ESN, but the new KNETI hash, and then re-establish the secure channel to the Connect(s)/RFS.

11.3.10. Compromised Key or Secret: nToken KNETI

Compromise Type

A brute force attack on KNETI encrypted blob held in KNETI file in the KMData folder.

Impact

KNETI is compromised and must not be used

Recovery Action

For every Connect that the affected client has communicated with, use the Front Panel or serial console to remove the client's configuration data.

For any RFS that the affected client has communicated with, update the RFS's configuration filer to remove the client's configuration data.

Manually delete the kneti file identified as `kneti-nToken ESN`.

- On Windows, it is stored in `C:\ProgramData\hardserver.d\`.
- On Linux, it is stored in `/opt/nfast/hardserver.d/`.

Reset the nToken.

Isolate client and investigate unauthorized access to KMData file and integrity of client.

Once resolved re-configure the Connects/RFS that this client communicated with using same client's IP address/ESN, but the new KNETI hash, and then re-establish the secure channel to the Connect(s)/RFS.

An attack on the client host system to disclose the client SSH private keys or recovering pri vate SSH client keys from backups

Impact

An attacker can access the HSM and run platform services using the compromised keys.

Recovery Action

Destroy the nShield Connect because its integrity can no longer be guaranteed. Configure a new nShield Connect to communicate with a client. Recreate the client and server HSM keys.

Factory reset the HSM to prevent access with the compromised client SSH keys. Isolate the client host system and investigate the unauthorized access to the client host system or backup. Re-enroll the client host system with the HSM to create new client SSH keys.

11.3.11. Compromised Key or Secret: Imported application keys

Application keys can also be imported into the HSM. Any secret/private application key imported in plaintext should be treated as potentially compromised if:

- The confidentiality and integrity of an imported secret/private key cannot be verified
- The provenance of the key is unknown (it has not come from a trusted party).

11.4. Deleting a Security World

You can re-initialise an HSM to use a new Security World if, for example, you believe that your existing Security World has been compromised. This must be done for all HSMs that hosted the old Security World, however:

- You are not able to access any keys that you previously used in a deleted Security World
- It is recommended that you reformat any standard nShield cards that were used as Operator Cards within this Security World before you delete it.



Except for nShield Remote Administration Cards, if you do not reformat the smart cards used as Operator Cards before you delete your Security World, you must throw them away because they cannot be used, erased, or reformatted without the old Security World key.

You must reformat, reuse or destroy the smart cards from a deleted Security World's ACS. If these cards are not overwritten or destroyed, then an attacker with these smart cards, a copy of your data (for example, a weekly backup) and access to any nShield HSM can access your old keys.

11.5. Module failure

If a module fails and cannot be factory reset then application keys protected by module keys and NVRAM keys are potentially vulnerable to attack. In this instance procedural and technical access controls should be deployed to protect the module until secure destruction of the module occurs as described in Decommission and Disposal.

11.6. Tamper incident

Physical Security provides guidance on the physical security controls available on the different nShield platforms and the procedural controls required to maintain those physical security controls across the product's lifecycle.

If a tamper incident is observed the guidance in Security Incident and Response should be followed to manage the incident. The investigation will determine the extent of the attack. Once an HSM has been confirmed as being tampered its integrity can no longer be assured and it should be decommissioned and disposed of — see Decommission and Disposal for more information.

However, there are two instances where it is possible to recover the module from a tamper event. These are:

- nShield Solo XC tamper events - see nShield Solo XC physical security controls for more information.
- nShield Connect lid is either open or closed - see Tamper event for guidance on how to investigate the tamper and the criteria required to recover from the tamper. The occurrence of the event should be recorded and recovery authorized in accordance with the Customer's Security Incident and Response Policy.

12. Decommission and Disposal

12.1. nShield Connect and nShield Solo

When an HSM reaches the end of its operational life it should be securely decommissioned and disposed of.

1. The Security Procedures in the Customer's Security Policy should describe the decommissioning process. To decommission the HSM, all secret information that is used to protect your Security World should be erased. See Remove modules and delete Security Worlds for details of the HSM factory reset procedure. If the Customer's Security Procedures have specific requirements concerning the erasing of application key material, these procedures should be performed before the factory reset is performed.



An HSM factory reset will erase all application key material. On nShield Solo+ or nShield Solo XC this means erasing the Security World (e.g. with the `initunit` command). For nShield Connect+, nShield Connect XC, nShield 5s and nShield 5c this additionally means a full factory reset to ensure all other information is removed, such as erasure of the KNETI key and any files loaded from the RFS from Connect and 5c, and erasure of any loaded CodeSafe applications on 5s and 5c.

2. If no further operational requirements exist for the HSM, Customer Security Procedures should describe the disposal process. The Customer Security Procedures concerning the transportation of the unit should be adhered to.
3. The customer may have a secure destruction policy for decommissioned assets. As long as all secret information that is used to protect your Security World has been erased there is no requirement to securely destroy the HSM as it has been returned to its factory state.
4. However if the HSM has malfunctioned in a way that it is not possible to determine whether secret information used to protect the Security World has been erased, that is, the possibility exists that secret information may still present in the HSM, then the customer must refer to their Security Procedures to determine if the HSM should be destroyed. One option here is to use a data destruction service offered by private companies who can destroy the equipment in accordance with approved standards and provide a certificate of data destruction. Customer Security Procedures should describe the destruction process that ensures that all HSM components that contains secrets are completely destroyed.

5. Entrust will accept the return of decommissioned HSMs for secure destruction.

12.1.1. Recycling and disposal information

For recycling and disposal guidance, see the nShield product's Warnings and Cautions documentation.

12.1.2. Security World

If the Security World resident on the decommissioned HSM is no longer required then the ACS and OCS cards should be erased by formatting them. Formatting cards using the nShield tools will zeroize any storage used for secrets.

- The ACS can only be erased by a different Security World, for example, a replacement Security World, or on an HSM with no Security World loaded. You can, and should, reuse the smart cards from a deleted Security Worlds ACS. If you do not reuse or destroy these cards, then an attacker with these smart cards, a copy of your data (for example, a weekly backup) and access to any nShield HSM can access your old keys.
- Legacy OCS cards can only be erased on the Security World that they were created for. Therefore, ensure that the OCS is erased as a final step before the HSM is decommissioned. The cards can then be used for a new Security World. Modern Remote Administration Ready cards can be erased forcibly even without access to their creating Security World by using the `--ignoreauth` option to `slotinfo --format`. Note that this option is only permitted when either there is no Security World loaded, or when not in a strict-FIPS world.
- Once the steps outlined above any keys that exist in backup data are no longer usable.

If a new Security World is not required uninstall the Security World software. However, we recommend that you do not uninstall the Security World software unless you are either certain it is no longer required, or you intend to upgrade it.

13. Abbreviations

Abbreviation	Description
ACL	Access Control List
ACS	Administrator Card Set
AES	Advanced Encryption Standard
API	Application Program Interface
CMVP	Cryptographic Module Validation Program
CPU	Central Processing Unit
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
ECDSA	Elliptic Curve Digital Signature Algorithm
eIDAS	Electronic Identification and Signature (Electronic Trust Services)
ESN	Electronic Serial Number
FIPS	Federal Information Processing Standard
HSM	Hardware Security Module
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
IPvX	Internet Protocol version (where X is the version number)
JCA/JCE CSP	Java Cryptography Architecture/ Java Cryptography Extension Cryptographic Service Provider
LED	Light Emitting Diode
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
NVRAM	Non-volatile random-access memory
OCS	Operator Card Set
OS	Operating System
PCI	Peripheral Component Interconnect
PSU	Power Supply Unit

Abbreviation	Description
RFS	Remote File System
RSA	Rivest Shamir Adelman
RTC	Real Time Clock
SAM	Security Assurance Mechanism
SEE	Secure Execution Engine
SIEM	Security Information and Event Management
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TVD	Trusted Verification Device
UI	User Interface
USB	Universal Serial Bus
VPN	Virtual Private Network
WEEE	Waste Electrical and Electronic Equipment