nShield Security World

Microsoft CNG v13.3 Reference Guide

05 April 2024

Table of Contents

1. Introduction

Cryptography API: Next Generation (CNG) is the successor to the Microsoft Crypto API (CAPI) and its long-term replacement. The Security World Software implementation of Microsoft CNG is supported on Microsoft Windows Server 2016 and later releases. The nShield CNG providers offer the benefits of hardware-based encryption accessed through the standard Microsoft API, and support the National Security Agency (NSA) classified Suite B algorithms.

Before using the nShield CNG providers, run the nShield CNG Configuration Wizard to:

- configure HSM Pool mode for CNG as required.
- create a new Security World or specify an existing Security World to use.
- register the nShield CNG providers.
- configure the nShield CNG providers as default CNG providers for specific tasks.

For additional information, see the Microsoft CNG documentation: http://msdn2.microsoft.com/en-us/library/aa376210.aspx.

This guide describes the Microsoft Cryptography API: Next Generation (CNG) toolkit supplied by Entrust Security to help developers write applications that use nShield modules.

This toolkit, like the application plug-ins supplied by Entrust, uses the Security World paradigm for key storage. For an introduction to Security Worlds, see the *User Guide*.

1.1. Read this guide if...

Read this guide if you want to build an application that uses an nShield key-management module to accelerate cryptographic operations and protect cryptographic keys through a standard interface rather than the full nCore API.

This guide assumes that you are familiar with the concept of the Security World, described in the User Guide. It is intended for experienced programmers and assumes that you are familiar with the following documentation:

- The *nCore Developer Tutorial*, which describes how to write applications using an nShield module.

- The *nCore API Documentation* (supplied as HTML), which describes the nCore API.

1.2. Model numbers

Model numbering conventions are used to distinguish different nShield hardware security devices. In the following table, *n* represents any single-digit integer.

Model number	Used for
NH2047	nShield Connect 6000
NH2040	nShield Connect 1500
NH2033	nShield Connect 500
NH2068	nShield Connect 6000+
NH2061	nShield Connect 1500+
NH2054	nShield Connect 500+
NH2075-B	nShield Connect XC Base
NH2075-M	nShield Connect XC Medium
NH2075-H	nShield Connect XC High
NH2075-B	nShield 5c Base
NH2075-M	nShield 5c Medium
NH2075-H	nShield 5c High
NH2082	nShield Connect XC SCAP
NH2089-B	nShield Connect XC Base - Serial Console
NH2089-M	nShield Connect XC Mid - Serial Console
NH2089-H	nShield Connect XC High - Serial Console
NH3003-B	nShield Connect CLX Base - Serial Console
NH3003-M	nShield Connect CLX Mid - Serial Console
NH3003-H	nShield Connect CLX High - Serial Console
nC2021E-000, NCE2023E-000	nToken PCIe
nC3*nnn*E-*nnn*, nC4*nnn*E-*nnn*	nShield Solo PCIe

Model number	Used for
nC30n5E-*nnn*, nC40n5E-*nnn*	nShield Solo XC PCIe
nC30*nn*U-10, nC40*nn*U-10	nShield Edge
NC5536E-B	nShield 5s Base
NC5536E-M	nShield 5s Medium
NC5536E-H	nShield 5s High

1.3. Security World Software default directories

The default locations for Security World Software and program data directories on English-language systems are summarized in the following table:

Directory Name	Environment Variable	Windows Server 2016	Linux
nShield Installation	`NFAST_HOME`	`C:\Program Files\nCipher\nfast`	`/opt/nfast/`
Key Management Data	`NFAST_KMDATA`	`C:\ProgramData\nCipher\Key Management Data`	`/opt/nfast/kmdata/`
Dynamic Feature Certificates	`NFAST_CERTDIR`	`C:\ProgramData\nCipher\Feature Certificates`	`/opt/nfast/femcerts/`
Static Feature Certificates		`C:\ProgramData\nCipher\Features`	`/opt/nfast/kmdata/features/`
Log Files	`NFAST_LOGDIR`	`C:\ProgramData\nCipher\Log Files`	`/opt/nfast/log/`

By default, the Windows `%NFAST_KMDATA%` directories are hidden directories. To see these directories and their contents, you must enable the display of hidden files and directories in the **View** settings of the **Folder Options**.

Dynamic feature certificates must be stored in the directory stated in the default directories table.

The directory shown for static feature certificates is an example location. You can store those certificates in any directory and provide the appropriate path when using the Feature Enable

> Tool. However, you must not store static feature certificates in the dynamic features certificates directory. For more information about feature certificates, see the *User Guide* for your HSM.

The absolute paths to the Security World Software installation directory and program data directories on Windows platforms are stored in the indicated nShield environment variables at the time of installation. If you are unsure of the location of any of these directories, check the path set in the environment variable.

The instructions in this guide refer to the locations of the software installation and program data directories by their names (for example, Key Management Data) or:

- For Windows, nShield environment variable names enclosed in percent signs (for example, `%NFAST_KMDATA%`).
- For Linux, absolute paths (for example, `/opt/nfast/kmdata/`).

`NFAST_KMDATA` cannot be a symbolic link.

If the software has been installed into a non-default location:

- For Windows, ensure that the associated nShield environment variables are re-set with the correct paths for your installation.
- For Linux, you must create a symbolic link from `/opt/nfast/` to the directory where the software is actually installed. For more information about creating symbolic links, see your operating system's documentation.

1.4. Utility help options

Unless noted, all the executable utilities provided in the `bin` subdirectory of your nShield installation have the following standard help options:

`-h|--help` displays help for the utility

`-v|--version` displays the version number of the utility

`-u|--usage` displays a brief usage summary for the utility.

1.5. Further information

This guide forms one part of the information and support provided by Entrust.

The *nCore API Documentation* is supplied as HTML files installed in the following locations:

- Windows:
  - API reference for host: `%NFAST_HOME%\document\ncore\html\index.html`
  - API reference for SEE: `%NFAST_HOME%\document\csddoc\html\index.html`
- Linux:
  - API reference for host: `/opt/nfast/document/ncore/html/index.html`
  - API reference for SEE: `/opt/nfast/document/csddoc/html/index.html`

The Java Generic Stub classes, nCipherKM JCA/JCE provider classes, and Java Key Management classes are supplied with HTML documentation in standard Javadoc format, which is installed in the appropriate `nfast\java` or `nfast/java` directory when you install these classes.

1.6. 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 recommended 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. Because of this, you should consider the procedures and actions required to upgrade devices in the field when deploying and maintaining your HSMs.

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

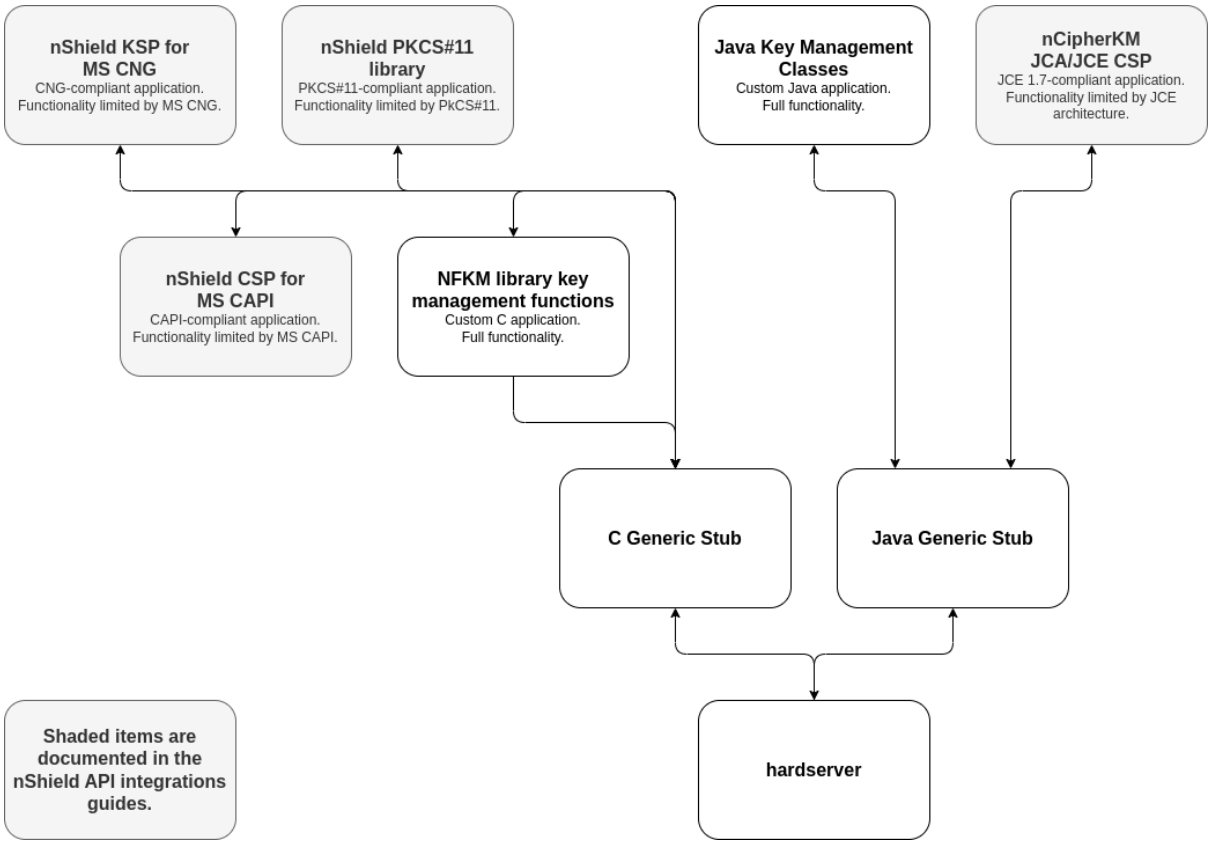
We recommend that you monitor the Announcements & Security Notices section on Entrust nShield, https://nshieldsupport.entrust.com, where any announcement of nShield Security Advisories will be made.

1.7. Contacting Entrust nShield Support

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

2. nShield Architecture

This chapter provides a brief overview of the Security World Software architecture. The following diagram provides a visual representation of nShield architecture and the documentation that relates to it.



2.1. Security World Software modules

nShield modules provide a secure environment to perform cryptographic functions. Key-management modules are fitted with a smart card interface that enables keys to be stored on removable tokens for extra security. nShield modules are available for PCI buses and also as network-attached Ethernet modules (nShield Connect).

2.2. Security World Software server

The Security World Software server, often referred to as the `hardserver`, accepts requests by means of an interprocess communication facility (for example, a domain socket on Linux or named pipes or TCP/IP sockets on Windows).

The Security World Software server receives requests from applications and

passes these to the nShield module(s). The module handles these requests and returns them to the server. The server ensures that the results are returned to the correct calling program.

You only need a single Security World Software server running on your host computer. This server can communicate with multiple applications and multiple nShield modules.

2.3. Stubs and interface libraries

An application can either handle its own cryptographic functions or it can use a cryptographic library:

- If the application uses a cryptographic library that is already able to communicate with the Security World Software server, then no further modification is necessary. The application can automatically make use of the nShield module.
- If the application uses a cryptographic library that has not been modified to be able to communicate with the Security World Software server, then either Entrust or the cryptographic library supplier need to create adaption function(s) and compile them into the cryptographic library. The application users then must relink their applications using the updated cryptographic library.

If the application performs its own cryptographic functions, you must create adaption function(s) that pass the cryptographic functions to the Security World Software server. You must identify each cryptographic function within the application and change it to call the nShield adaption function, which in turn calls the generic stub. If the cryptographic functions are provided by means of a DLL or shared library, the library file can be changed. Otherwise, the application itself must be recompiled.

2.4. Using an interface library

Entrust supplies the following interface libraries:

- Microsoft Cryptography API: Next Generation (CNG)
- Microsoft CryptoAPI (CAPI)
- PKCS #11
- nCipherKM JCA/JCE CSP

Third-party vendors may supply nShield-aware versions of their cryptographic libraries.

The functionality provided by these libraries is the intersection of the functionality provided by the nCore API and the functionality provided by the standard for that library.

Most standard libraries offer fewer key-management options than are available in the nCore API. However, the nShield libraries do not include any extensions to their standards. If you want to make use of features of the nCore API that are not offered in the standard, you should convert your application to work directly with the generic stub.

On the other hand, many standard libraries include functions that are not supported on the nShield module, such as support for IDEA or Skipjack. If you require a feature that is not supported on the nShield module, contact Support because it may be possible to add the feature in a future release. However, in many cases, features are not present on the module for licensing reasons, as opposed to technical reasons, and Entrust cannot offer them in the interface library.

2.5. Writing a custom application

If you choose not to use one of the interface libraries, you must write a custom application. This gives you access to all the features of the nCore API. For this purpose, Entrust provides generic stub libraries for C and Java. If you want to use a language other than C orJava, you must write your own wrapper functions in your chosen programming language that call the C generic stub functions.

Entrust supplies several utility functions to help you write your application.

2.6. Acceleration-only or key management

You must also decide whether you want to use key management or whether you are writing an acceleration-only application.

Acceleration-only applications are much simpler to write but do not offer any security benefits.

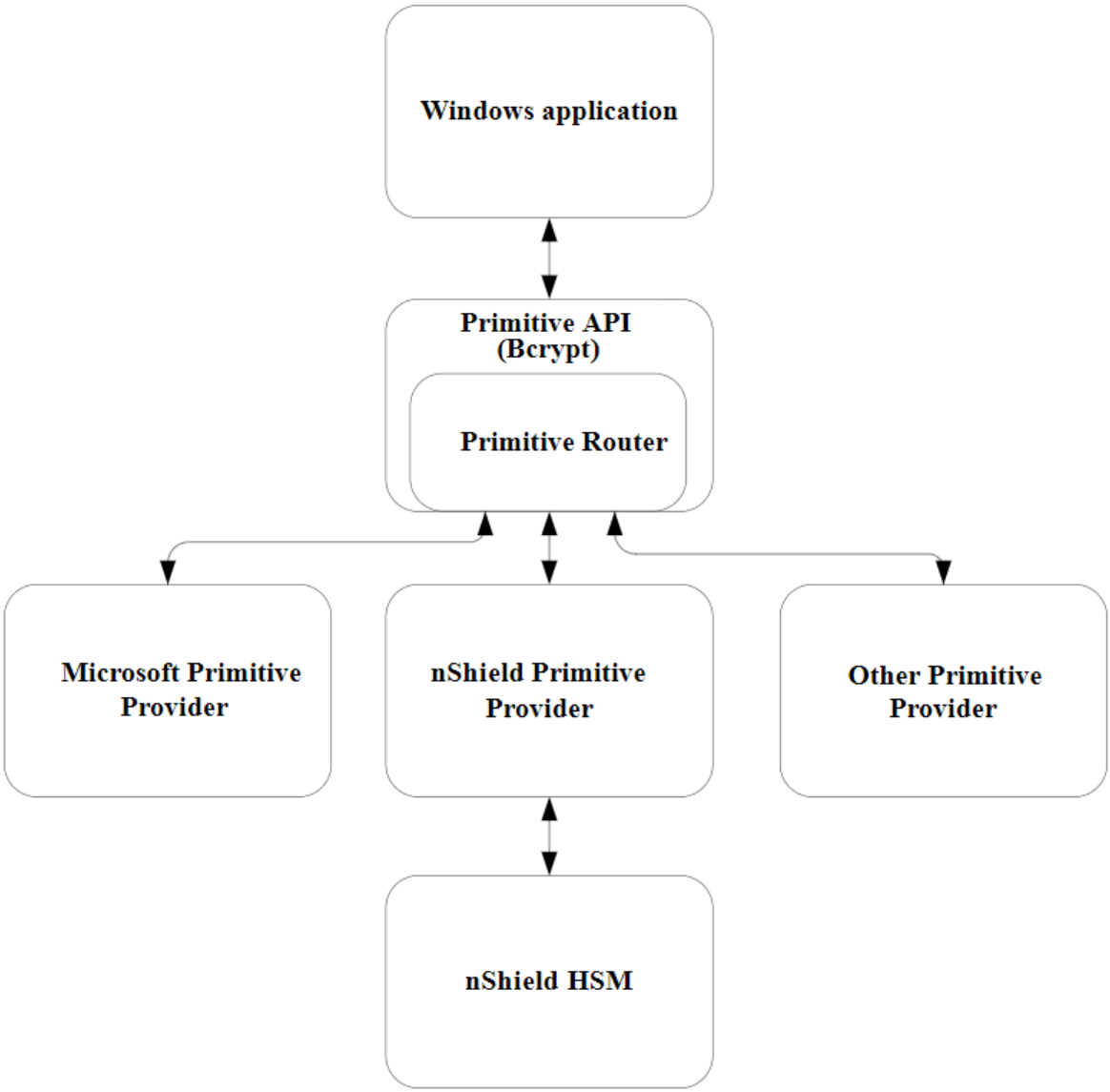
The Microsoft CryptoAPI, Java JCE, PKCS #11, as well as the application plug-ins, use the Security World paradigm for key storage.

If you are writing a custom application, you have the option of using the Security World mechanisms, in which case your users can use either KeySafe or the command-line utilities supplied with the module for many key-management operations. This means you do not have to write these functions yourself.

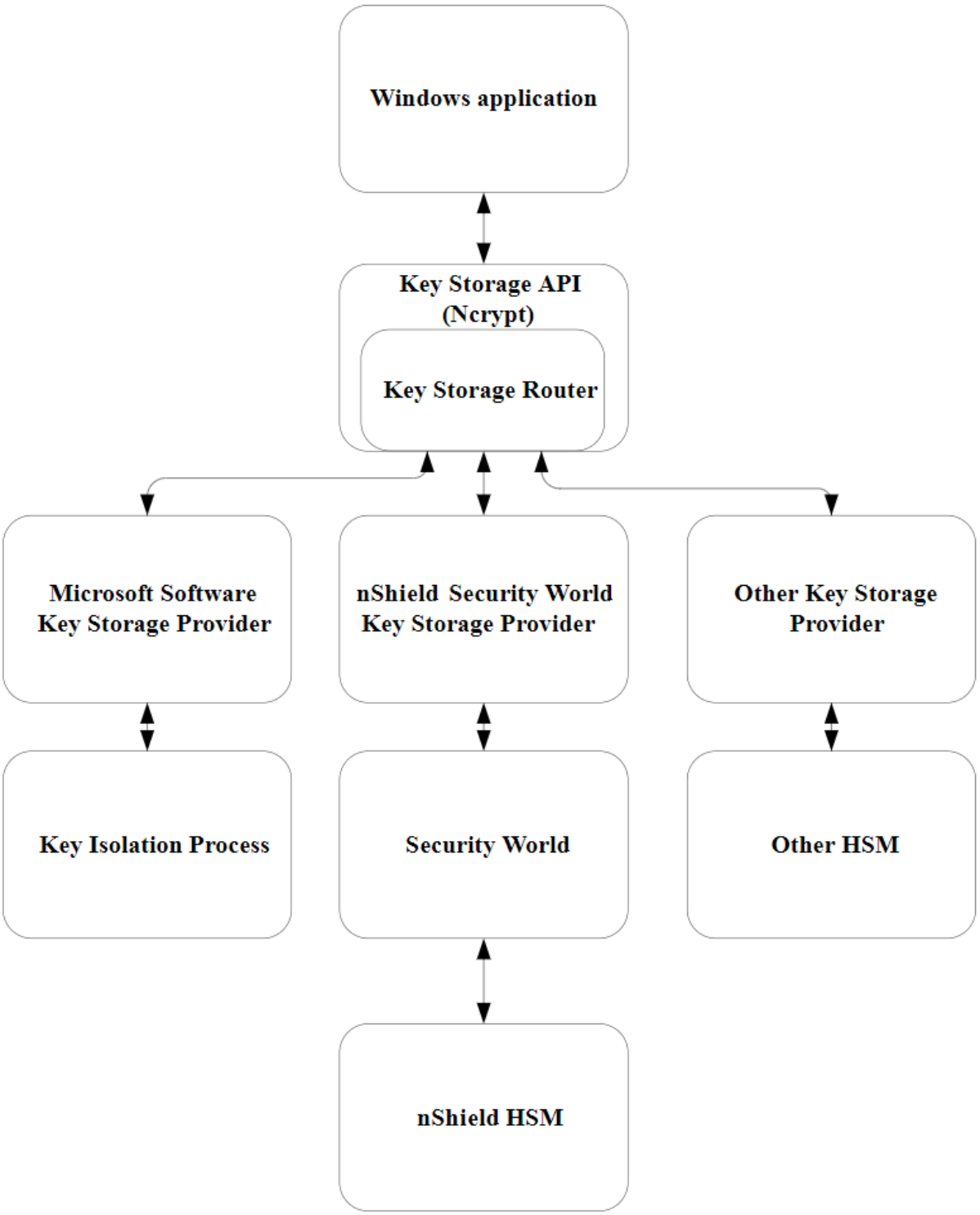
The NFKM library gives you access to all the Security World functionality.

3. CNG Architecture Overview

CNG handles cryptographic primitives and key storage through separate APIs. In both cases a Windows application contacts a router, which forwards the cryptographic operation to the provider that is configured to handle the request. For an illustration of communication between the architecture layers for cryptographic primitives, see the following diagram.



For an illustration of communication between the architecture layers for cryptographic key storage, see the following diagram.


Windows application
Key Storage API (Ncrypt)
Key Storage Router
Microsoft Software Key Storage Provider
nShield Security World Key Storage Provider
Other Key Storage Provider
Key Isolation Process
Security World
Other HSM
nShield HSM

4. Supported Algorithms

This chapter lists the National Security Agency (NSA) classified Suite B algorithms supported by the nShield CNG providers.

The MQV algorithm is not supported by the nShield CNG providers.

Some mechanisms may be restricted from use in Security Worlds conforming to FIPS 140 Level 3. See the *User Guide* for your HSM for more information.

4.1. Signature interfaces (key signing)

Interface name	*Type of support*
RSA PKCS#1 v1	Hardware
RSA PSS	
DSA	
ECDSA_P224	
ECDSA_P256	
ECDSA_P384	
ECDSA_P521	

Hashes used with ECDSA must be of the same length or shorter than the curve itself. If you attempt to use a hash longer than the curve the operation returns `NOT_SUPPORTED`. In FIPS 140 Level 3 Security Worlds, curves must be of an approved type and length.

4.2. Hashes

Hash name	*Type of support*
SHA1	Hardware (HMAC only)/software
SHA256	
SHA384	
SHA512	
SHA224	Hardware (HMAC only, requires firmware version 2.33.60 or later)/software
MD5	Hardware (HMAC only)/software

4.3. Asymmetric encryption

Algorithm name	*Type of support*
RSA Raw (NCRYPT_NO_PADDING_FLAG)	Hardware
RSA PKCS#1 v1 (NCRYPT_PAD_PKCS1_FLAG)	
RSA OAEP (NCRYPT_PAD_OAEP_FLAG)	

4.4. Symmetric encryption

Algorithm name	*Type of support*
RC4	Hardware and Software
AES ECB,CBC	
DES ECB,CBC	
3DES ECB,CBC	
3DES_112 ECB,CBC	

4.5. Key exchange

Protocol name	*Type of support*
DH	Hardware
ECDH_P224	
ECDH_P256	
ECDH_P348	
ECDH_P521	

Elliptic curve cryptography algorithms must be enabled before use. Use the `fet` command-line utility with an appropriate certificate to enable a purchased feature. If you enable the elliptic curve feature on your modules after you first register the CNG providers, you must run the configuration wizard again for the elliptic curve algorithm providers to be registered. For more information about registering the CNG providers, see the *User Guide* for your HSM.

4.6. Random Number Generation

Name	*Type of support*
RNG	Hardware

5. Key Authorization

When an application needs keys that are protected by an Operator Card Set or a Softcard, a user interface is invoked to prompt the application user to insert the smart card and/or enter appropriate passphrases.

The user interface prompt is not provided if your application is working in silent mode. The nShield CNG providers attempt to load the required authorization (for example, from an Operator Card that has already been inserted) but fail if no authorization can be found. For more information about silent mode, refer to the documentation of the CNG Key Storage Functions at: http://msdn2.microsoft.com/en-us/library/aa376208.aspx.

When the CNG application is running in Session 0 (that is, loaded by a Windows service), the user interface is provided by an agent process **nShield Service Agent** that is started when the user logs in. This agent, when running, is shown in the Windows System Tray. All user interaction requests from a CNG application running in Session 0 cause dialogs to be raised by the agent allowing the user to select cardsets, modules and enter passphrases. The interaction with the user is functionally identical to that described in this section.

There can only be one instance of the agent running (indicated by a blue globe in the Tray Notification area in the toolbar). Attempts to start a second instance will fail with a **CreateNamedPipe** error. If the agent is not running, attempts to invoke dialogs through it will fail and this is logged in the Windows Event Log. It can be restarted by logging off and on or by explicitly executing either `%NFAST_HOME%\bin\nShield_service_agent64.exe` or `%NFAST_HOME%\bin\nShield_service_agent.exe`. On 64 bit platforms either of these can be used irrespective of the bit size of the underlying application.

For more information about autoloadable Card Sets and the considerations of silent mode, see the authorisation requests diagram towards the end of this section.

You define key protection and authorization settings with the CNG Configuration Wizard on the **Key Protection Setup** screen. For more information about the CNG Configuration Wizard, see the *User Guide* for your HSM.

The options on this screen that are relevant to key protection and authorization

are:

- **Module protection**

  Select this option to make keys module protected by default.

- **Softcard Protection**

  Select this option to generate new keys with a particular Softcard by default.

- **Operator Card Set protection**

  Select this option to generate new keys with a particular Operator Card Set by default.

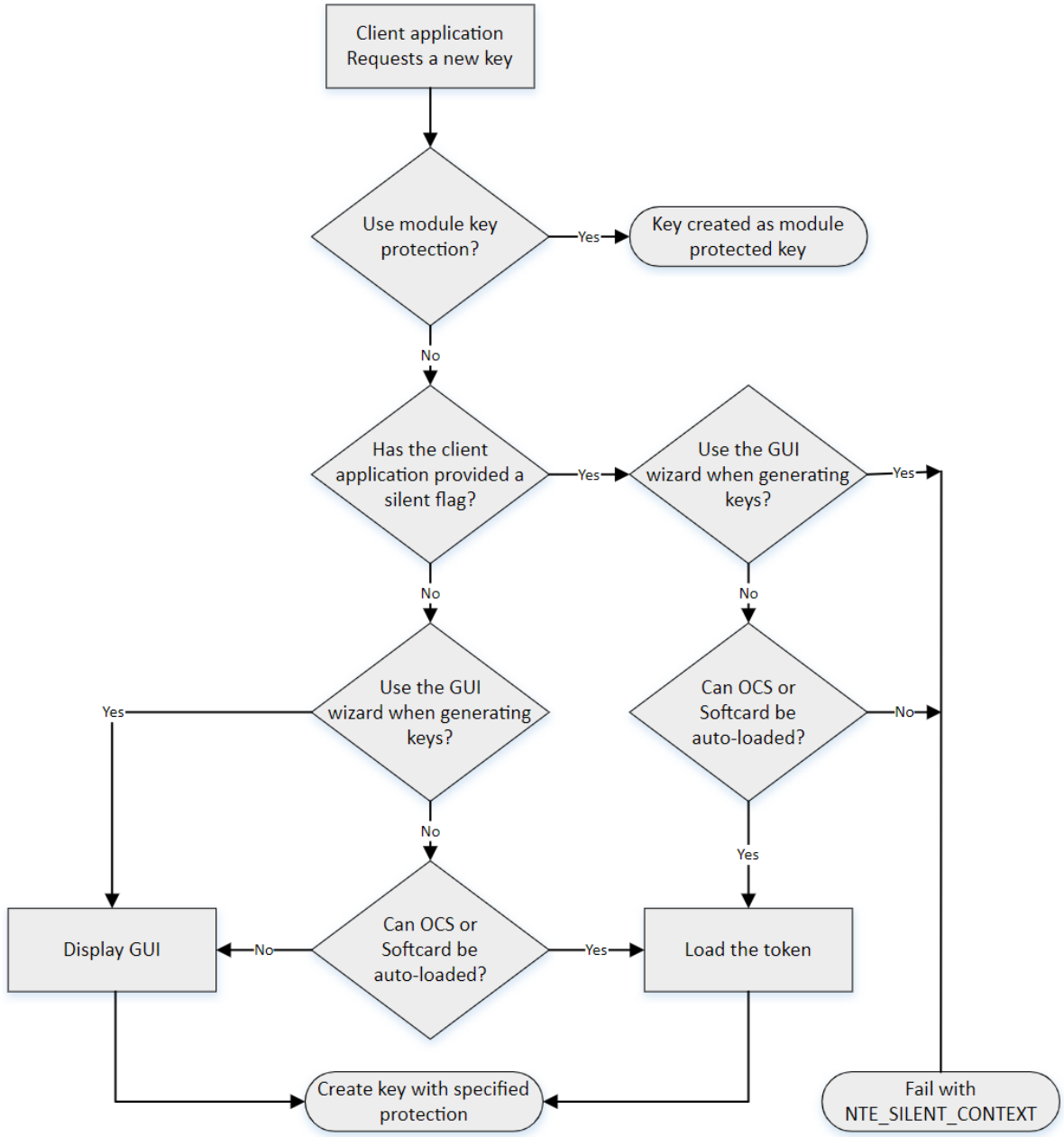
- **Allow any protection method to be selected in the GUI when generating**

  Select this option to defer selection of the key protection until the key is generated. When generating a key, the choice between Module protection, or protection with an existing Softcard or Operator Card Set, will be offered.

If you select Softcard or Operator Card Set protection, you will be offered the choice between selecting an existing protection token and creating a new one on the next page.

The CNG Configuration Wizard can be re-run to change the default protection. Existing keys that were generated with a different protection can still be loaded even if they don't match the protection that was selected in the wizard.

The nShield GUI is never enabled for calls with a valid **Silent** option. If the **Use the GUI wizard..** option is selected, and the providers have been passed the **Silent** option, key generation will always fail. For Softcard and Operator Card Set protection, **Silent** mode will work only if the Softcard or Operator Card Set can be autoloaded without prompting for user interaction or passphrase entry.



FIPS 140 Level 3 environments always require card authorization for key creation. When using the CNG Primitive Functions the user is not prompted to provide card authorization, but the request fails if no card is provided.

The key storage providers always respect calls made with the `Silent` option. Primitive providers never display a user interface.

Applications may have a mechanism to disable silent mode operation, thereby allowing appropriate passphrases to be entered. Ensure that you configure applications to use an appropriate level of key protection. For example, in Microsoft Certificate Services, you must select the **Use strong private key**

protection features provided by the CSP option to disable silent mode operation.

6. Key Use Counting

You can configure the CNG provider to count the number of times a key is used. Use this functionality, for example, to retire a key after a set number of uses, or for auditing purposes.

Key counting is not supported in HSM Pool mode.

To enable key use counting in the Security World Key Storage Provider, call `NCryptSetProperty` with `NCRYPT_USE_COUNT_ENABLED_PROPERTY` on the provider handle. Alternatively, to override the behavior of third-party software that would not otherwise provide the user with the option to enable key use counting, use one of the following methods:

- Set the environment variable `NCCNG_USE_COUNT_ENABLED` to `1`.
- Set the registry key `Software\nCipher\CryptoNG\UseCountEnabled` to `1`.

Keys created while the provider has key use counting enabled continue to have their use counts incremented, regardless of the state of the provider's handle. Key use counts are not recorded for keys created while the `NCRYPT_USE_COUNT_ENABLED_PROPERTY` is disabled on the provider handle.

Because the key counter is a 64-bit area in a specific module's NVRAM, the counted keys are specific to a single module. When a key is created you are prompted to specify which module to use, unless there is only one module in the Security World, or `preload` was used to preload authorization from an ACS on only one module.

The key counter is incremented each time a private key is used to:

- sign
- decrypt
- negotiate a secret agreement.

To test the performance of keys with counters, run the `cngsoak` command with the `-C` option:

```
cngsoak -C --sign --length=1024
```

To view the current key use count for keys, run the `cnglist` command with the `--list-keys` and `--verbose` options:

```
cnglist --list-keys --verbose
```

7. Using CAPI Keys

We now provide the capability to use keys generated by CAPI in CNG applications. This is provided through the standard `NCryptOpenKey` CNG API call. Passing either `AT_SIGNATURE` or `AT_KEYEXCHANGE` as the `dwLegacyKeySpec` parameter and the CAPI container name as the `pszKeyName` parameter will invoke this mode of operation. The CAPI key will be loaded into the CNG provider and will behave as if it was a CNG key. Any key authorization required will be handled with a user interface being invoked to prompt the application user to insert the smart card or enter appropriate passphrases. There is support for Key Usage and Key Counting properties.

The CNG application has to be written such that it calls `NCryptOpenKey` to open a CAPI key explicitly.

8. Utilities

Use the `nfkmverify` command-line utility to check the security of all stored keys in the Security World. Use `nfkminfo`, `nfkmcheck`, and other command-line utilities to assist in this process. For more information about these command-line utilities, see the *User Guide* for your HSM.

The following table lists the utilities specific to the nShield CNG CSP:

x86	x64	Utility description
`cngimport32.exe`	`cngimport.exe`	This key migration utility is used to migrate Security World, CAPI, and CNG keys to the Security World Key Storage Provider.
`cnginstall32.exe`	`cnginstall.exe`	This utility is the nShield CNG CSP installer. Only use this utility to remove or reinstall the provider DLLs and associated registry entries manually.
`cnglist32.exe`	`cnglist.exe`	This utility lists information about CNG CSP.
`cngregister32.exe`	`cngregister.exe`	This is the nShield CNG CSP registration utility. You can use it to unregister and re-register the nShield providers manually.
`ncsvcdep32.exe`	`ncsvcdep.exe`	This utility is the service dependency tool. You can configure some service based applications, such as Microsoft Certificate Services and IIS, to use the nShield CNG CSP. The nShield Service dependency tool allows you to add the nFast Server to the dependency list of such services.
`configure-csp-poolmode32`	`configure-csp-poolmode`	This utility allows you to configure HSM Pool mode for the nShield CNG CSP without using the CNG wizard.

For more information about the command-line utilities, see the *User Guide* for your HSM.

9. Environment Variables for CNG Protection

A set of environment variables are supported for controlling CNG protection options on a per-application basis. These variables are documented here to facilitate more complicated deployments, but it should be noted that they are liable to change between releases.

Environment Variable	Description
`NCCNG_PIN`	Passphrase for Softcard. This enables the passphrase to be specified programmatically rather than through the GUI passphrase prompt. Note: This can expose your passphrase.  It is recommended that this be set in a context where the passphrase will be visible only to the user or service that should have access to this passphrase. It should **not** be set as a machine-wide environment variable.
`NCCNG_USE_MODULE_KEYS`	• If set to 1, module protection will be used for new keys that are generated. • If set to 0, the `NCCNG_PROTECTION_TOKEN` environment variable controls the protection option used.
`NCCNG_PROTECTION_TOKEN`	If `NCCNG_USE_MODULE_KEYS` is set to 0 (or a protection option other than module key protection or HSM pool mode was selected in the wizard) this environment variable enables the protection token to be specified for new keys that are generated. • If set to `softcard:HASH` the Softcard with the specified hash will be used. • If set to `cardset:HASH` the OCS with the specified hash will be used. • If set to anything else (e.g. wizard), the GUI key protection wizard will be used. The HASH for Softcard or OCS protections refers to its Security World hash in hexadecimal, which can be identified using `nfkminfo -s` for softcards or `nfkminfo -c` for OCS.

Environment Variable	Description
NCCNG_ALWAYS_USE_AGENT	By default, if a CNG provider must display GUI, it will display it in the calling application if not in Session 0, and in the nShield Service Agent if running in Session 0 (e.g. running as a service). Setting NCCNG_ALWAYS_USE_AGENT to 1 forces CNG GUI prompts to always be displayed in the nShield Service Agent regardless of whether it is running in Session 0. (If setting this value to 1 ensure that the nShield Service Agent is running).