nShield Security World

Microsoft CryptoAPI Guide for nShield Security World v13.6.8

17 February 2025

Table of Contents

1. Introduction

Entrust provides a Cryptographic Service Provider (CSP) that implements the Crypto API (CAPI) supported in Windows 2008 and later.

Except where this document specifies otherwise, the Security World Software implementation conforms to the Microsoft CSP interface. For more information, see the Microsoft CSP documentation.

This guide describes the Microsoft CryptoAPI (MSCAPI) 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 nShield Security World v13.6.8 Management 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. 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. Additional useful documentation

Refer to nShield Security World v13.6.8 Management Guide and nShield v13.6.8 HSM User Guide for additional information about Security Worlds and nShield HSMs.

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 Optional features.

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 Manage ment 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 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. 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.

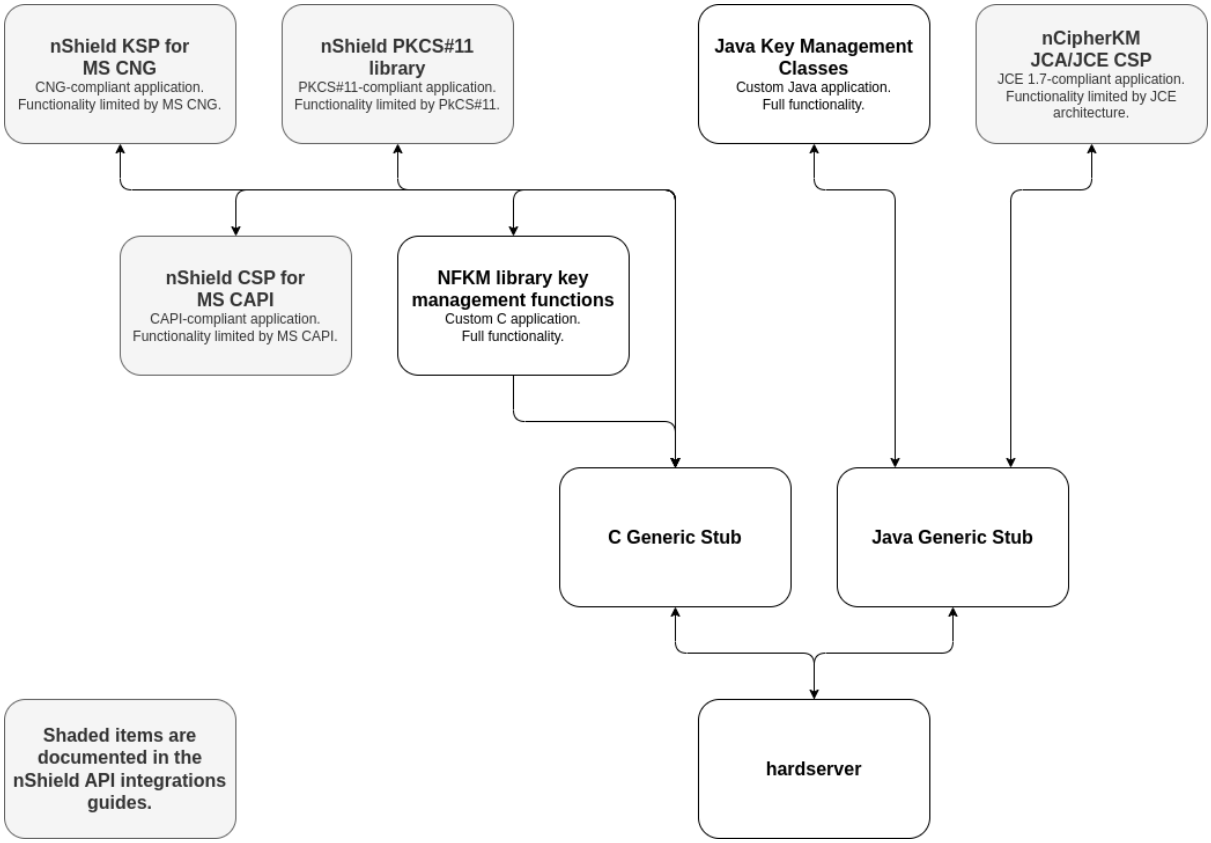
We recommend that you monitor the Announcements & Security Notices section on Entrust nShield, https://nshieldsupport.entrust.com, where any announcement of nShield Security Advisories will be made.

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 sup ported on the nShield module, contact Support because it may be possible to add the fea- ture 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 bene fits.

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. Microsoft CAPI CSP

The following provider types are supported:

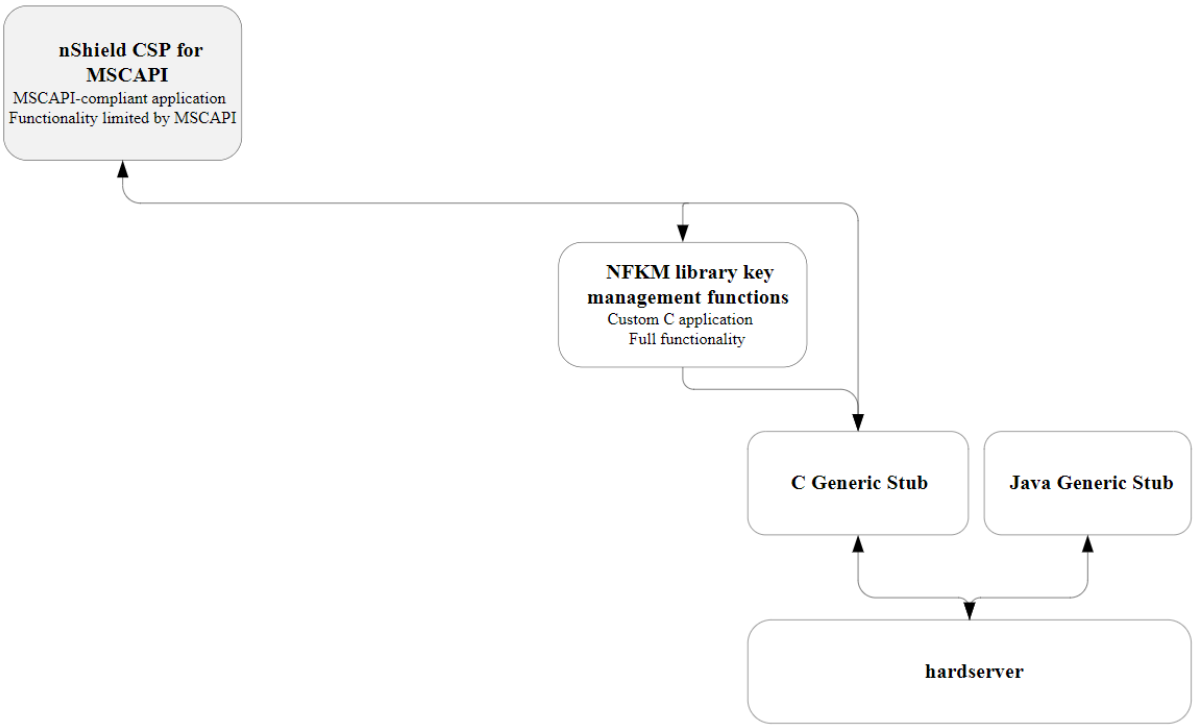
- `PROV_RSA_FULL` (nShield Enhanced Cryptographic Provider)
- `PROV_RSA_AES` (nShield Enhanced RSA and AES Cryptographic Provider)
- `PROV_RSA_SCHANNEL` (nShield Enhanced SChannel Cryptographic Provider)
- `PROV_DSS` (nShield DSS Signature Cryptographic Provider)
- `PROV_DSS_DH` (nShield Enhanced DSS and Diffie-Hellman Cryptographic Provider)
- `PROV_DH_SCHANNEL` (nShield Enhanced DSS and Diffie-Hellman SChannel Cryptographic Provider)

We also provide a modulo exponentiation offload DLL that enables the Microsoft CSP to take advantage of the computational power of an nShield module without added security benefits. This is useful for interoperation with applications that do not allow the user to choose the CSP.

Unlike the Microsoft CSPs, the nShield CSPs do not support the exporting of private keys.

You should not need to make any adjustments to your code in order to use the nShield CSPs. However, the nShield module is an asynchronous device capable of performing several operations at once. To achieve maximum performance from the module, structure your application in a multithreaded manner so that it can make several simultaneous requests to the CSP.

The following diagram illustrates how the Microsoft CryptoAPI interface works with the nShield APIs.


nShield CSP for MSCAPI
MSCAPI-compliant application
Functionality limited by MSCAPI
NFKM library key management functions
Custom C application
Full functionality
C Generic Stub
Java Generic Stub
hardserver

4. CSP Setup and Utilities

4.1. Install the CAPI CSP

Entrust provides a CSP installation wizard that enables you to install the CAPI CSP. A short-cut to the CSP installation wizard is created in the **Start** menu when the Security World Software is installed: **Start > Entrust nShield Security World**. The CSP installation wizard registers the CAPI CSP as a key provider on your system.

Install the CAPI CSP using the 32-bit or the 64-bit installation wizard depending on whether you want to run 32-bit or 64-bit applications with the nShield CAPI provider.

4.2. Additional functionality of the CSP installation wizard

You can also perform the following actions with the CSP installation wizard:

- Load existing Security World
- Set up the modexp offload DLL
- Generate new Operator Card Sets (OCS)
- Configure the setup parameters of the CSP, including HSM Pool mode.
  With module firmware version 2.65.2 or later, if your application only uses module protected keys, you can use HSM Pool mode with multiple hardware security modules. HSM Pool mode exposes a single pool of HSMs and supports returning or adding a hardware security module to the pool without restarting the system. With a FIPS 140 Level 3 Security World, keys cannot be created in HSM Pool mode, however keys created outside HSM Pool mode can be used in HSM Pool mode.

The CSP installation wizard is not suitable for creating Security Worlds. Use new-world and createocs to create your Security World.

The standard Security World utility nfkmverify should be used to check the security of all stored keys in the Security World; nfkminfo, nfkmcheck and other standard utilities can also be used to assist in this process.

4.3. Uninstall the CAPI CSP

To uninstall the CAPI CSP and unregister it as a cryptographic provider on your system, run the `cngregister` and `cnginstall` commands with the `-U` option. For more information, see Utilities for the CAPI CSP.

5. Utilities for the CAPI CSP

CSP version 1.11.0 and later provides you with the following utilities. These utilities can help you migrate from the older Windows registry-based CSP container storage to the newer CSP format, There are also utilities to manage the interfaces between the MSCAPI library and the module. The CSP format stores all information about a Security World in the Key Management Data directory.

Utility	Description
`cspcheck`	This utility checks that CSP container files are intact and uncorrupted, and also that referenced key files exist. Use `cspcheck` in conjunction with `nfkmcheck`, but run `nfkmcheck` first in order to test the integrity of your Security World files.
`cspimport`	This utility allows you to insert keys manually into existing CSP containers. This utility has two modes that either allow you to change a container's key association to that of an arbitrary Security World key or to copy CSP keys between containers.
`cspmigrate`	This utility moves the CSP container information from the registry into the Security World. If a new container already exists and has a key in it, and an iden tically-named old container exists with the same key, the utility asks you which key to keep. You can either: Enter `-q` to keep the new keys. Enter `-f` to overwrite new keys with old keys.
`cspnvfix`	Regenerate an erased NVRAM key counter area for a specified nShield CSP key.
`csptest`	Test the installed Cryptographic Service Providers. This can be used to list the capabilities of installed nShield and Microsoft CSPs or to perform a soak test.
`csputils`	This utility lists CSP containers and provides detailed information about them, including the keys present and the values of the counters for key-counted keys. It can also be used to delete container files if the current user has admin- istrative privileges.
`configure-csp-poolmode`	The `--mscapi` option allows HSM Pool mode to be enabled or disabled for the nShield CAPI CSP without using the CSP wizard.
`keytst`	This utility displays information about existing CSP key containers by using the Microsoft CryptoAPI. If you have the appropriate permissions, `keytst` also allows you to create and delete containers and their keys.

Each of these commands has an `-h` option that displays the usage mes-

sage for the command.

6. Supported Algorithms

The nShield CSPs support a similar range of algorithms to the Microsoft CSP.

6.1. Symmetric algorithms

- CALG_DES
- CALG_3DES_112 (double-DES)
- CALG_3DES
- CALG_RC4
- CALG_AES_128
- CALG_AES_192
- CALG_AES_256

6.2. Asymmetric algorithms

- CALC_RSA_SIGN (only Enhanced RSA and AES Cryptographic Provider)
- CALC_RSA_KEYX (only Enhanced RSA and AES Cryptographic Provider)
- CALC_DSA_SIGN (only Enhanced DSS and Diffie-Hellman Cryptographic Provider and DSS Signature Cryptographic Provider)
- CALC_DSS_SIGN (only Enhanced DSS and Diffie-Hellman Cryptographic Provider)
- CALC_DH_KEYX (only Enhanced DSS and Diffie-Hellman Cryptographic Provider)
- CALC_DH_SF (only Enhanced DSS and Diffie-Hellman Cryptographic Provider)
- CALC_DH_EPHEM (only Enhanced DSS and Diffie-Hellman Cryptographic Provider)

6.3. Hash algorithms

- CALG_SHA1
- CALG_SHA256
- CALG_SHA384
- CALG_SHA512
- CALG_SSL3_SHAMD5
- CALG_MD5
- CALG_MAC
- CALG_HMAC

In addition, the Enhanced SChannel Cryptographic Provider and the Enhanced DSS and Diffie-Hellman SChannel Cryptographic Provider support all the internal algorithm types necessary for SSL3 and TLS1 support.

The nShield CSPs do not support SSL2.

7. Key Generation and Storage

The nShield CSP generates public/private key pairs (RSA, DSA, and Diffie-Hellman keys) in the module. The keys are stored in the Security World as protected by key blobs. (For more information about Security Worlds, see nShield Security World v13.6.8 Management Guide). Natively generated keys have `mscapi` as the `appname` and the hash of the key as the `ident`.

As in the Microsoft CSP, up to two keys are allowed for each container. Containers themselves are stored as opaque data in the Security World. Containers contain no key information but serve to associate NFKM keys with CSP containers, as well as storing other miscellaneous information. They have `mscapi` as the `appname` and `container-`containerID as the `ident`, where *containerID* is calculated from a combination of the CSP name, the user's unique SID and the container name.

The default permissions on new containers created by the nShield CSP have changed in order to solve a problem with IIS version 6: in this version of IIS it was possible to create containers with an empty ACL, such that they were completely inaccessible.

The previous default container permissions came from the inherited permissions on the `NFAST_KMLOCAL` directory, and had no non-inherited permissions. The default Security World Software installation gives everyone full control of the `NFAST_KMLOCAL` directory.

The current software sets an explicit ACL on new containers created by the CSP but does not alter permissions on previously created containers. The new permissions are as follows:

- `READ` access for `EVERYONE`.
- `FULL` access for `BUILTIN\Administrators`.
- For user containers: `FULL` access for the current user.
- For machine containers: `FULL` access for `LOCALSYSTEM`.

No action is required on the user's part to invoke the new behavior.

Symmetric keys in the nShield CSP are generated and stored entirely in software. These keys are not hardware protected and are no more secure than the corresponding keys in the Microsoft CSP.

The values of the `KP_PERMISSIONS` flags for hardware protected keys are enforced in software, except for `CRYPT_EXPORTABLE` which is ignored.

All CSP-generated, hardware-protected keys have ACLs that allow both signing and encryption. Hardware-protected keys that have been generated by the CSP are never exportable

by the CSP; `CryptExportKey` always fails with a permissions error when called on such a key.

Container files and their associated key files can be moved freely between machines, as long as the user's SID is also valid on the destination machine. This is the case if the user in question is a domain user and both machines are on that domain. If the user's SID is not valid on the destination machine and keys are required to be shared between multiple machines, then the `cspimport` utility must be used to reassociate the Security World key file with the required destination container.

8. User Interface Issues

The nShield CSP supports hardware keys protected by either the module itself or by OCSs. Protecting keys with OCSs raises some user interface issues because the user interface needs to be displayed both at key-creation time and at key-loading time.

The choice of using module-protected keys or keys protected by OCSs is made in the install wizard. If, however, you generate keys protected by OCSs and then switch to module protection, then in most cases the keys protected by OCSs still require the user interface to be displayed in order to load them.

At key-generation time, if the `always display UI at key gen` flag is unset and an automatic Operator Card is present, the CSP uses the Card Set to protect the key, loading the shares automatically on all modules that contain a suitable card. (The flag is set using the install wizard.) Otherwise the CSP displays the user interface and blocks until the user interface is completed.

At key-loading time, if the key is protected by an automatic OCS, and the Card Set is present, then the key is loaded on all modules that contain a suitable card. Otherwise, the CSP displays the user interface and blocks until the user interface is completed; this requires the same steps as for key generation except for choosing the Card Set.

An automatic OCS means a card from a 1/*N* Card Set that is not protected by a passphrase. At either time, the user interface is completed when the user has chosen a Card Set and the modules on which to load the key and has performed the card and passphrase operations.

The CSP requires authorization to import keys (including public keys) and to generate keys when you have initialized your modules in the mode compatible with FIPS 140 Level 3. This means that you must have a card from your current Security World in the slot when you attempt any of these operations, even if you are generating a module-protected key. If a card is not present, the operation blocks, and the CSP displays a user interface that prompts you to insert a card.

The CSP honors the `CRYPT_SILENT` flag to `CryptAcquireContext`. If this flag is passed in and the CSP would otherwise have to put up the user interface for any of the reasons in the two previous paragraphs, it fails with the appropriate error message.

If the CSP is being loaded from a service process (for example when used from within IIS or the main Certificate Authority process), then that process does not necessarily have access to the user's desktop. This means that any UI displayed by the CSP may not appear on an attended desktop (or at all), and the underlying operation may well time out.

If this is the case (and you are not using the `CRYPT_SILENT` flag, for whatever reason), we rec

ommend that either you do not use OCS-protected keys or you use an automatic Card Set, so that the CSP does not display the UI.

9. Key Counting

The nShield CSP supports the PP_CRYPT_COUNT_KEY_USE parameter to CryptAcquireContext as long as the module with NVRAM is attached. Setting this parameter to a nonzero value causes all keys generated from that point to have nonvolatile use counters. The counter per sists until CryptReleaseContext is called or until the PP_CRYPT_COUNT_KEY_USE parameter is reset to 0.

Key counting is not directly supported by end-user applications such as IIS. It is only supported by Microsoft Certificate Services under Windows 2003 and later. However, it is possible to create a certificate that uses a key counter in cases where key counting is not directly supported.

Key counting is not supported in HSM Pool mode.

Keys that have counters can only be loaded on one module at a time. The key-generation and key-loading functions enforce this behavior. When you generate these keys, you must present your Administrator Cards in order to authorize the creation of the new NVRAM area.

You must not insert your Administrator Cards in an untrusted host.

To minimize the exposure of the Security Officer root key (K_{NSO}) when you generate a key with key counting enabled, you should create the Security World with an NVRAM delegation key that requires the presentation of fewer Administrative Cards than are required to load K_{NSO}.

If you reinitialize your module for any reason, all the NVRAM areas on that module are erased. You must then use cspnvfix to recreate the NVRAM areas for all the keys that have counters.

10. NVRAM-Stored Keys

The nShield CSP now supports creating keys protected by the module NVRAM. The `PP_NO_HOST_STORAGE` parameter to `CryptAcquireContext` is supported as long as the module with NVRAM is attached. Setting this parameter to a nonzero value causes all keys generated from that point to be generated with blobs in NVRAM. The counter persists until CryptReleaseContext is called or until the `PP_NO_HOST_STORAGE` parameter is reset to `0`.

The method of creating NVRAM-stored keys is very similar to the method of creating keys with NVRAM counters:

1. call `CryptAcquireContext` to get a handle to a container.
2. call `CryptSetProvParam` and set the `PP_NO_HOST_STORAGE` property to a non-zero value.

This causes any keys generated with that container handle to be generated with blobs in NVRAM until either of the following occurs:

- `CryptReleaseContext` is called with that container handle.
- `CryptSetProvParam` is called to set `PP_NO_HOST_STORAGE` to zero.

Creating NVRAM-stored keys requires insertion of the ACS quorum for NVRAM, in the same way as creating key counted keys.

`PP_NO_HOST_STORAGE` is a new value and will be set in the `wincrypt.h` header file in future versions of the Microsoft Platform SDK. The following example code can be used until then to define the value correctly:

```
#ifndef PP_NO_HOST_STORAGE
#define PP_NO_HOST_STORAGE 44
#endif
```

This feature is only available to users writing CAPI code directly. To use an NVRAM-stored key in a client application (for example IIS or the Microsoft Certificate Authority), first create the key with the `keytst` command-line tool, and then transfer the key across to the required container with the `cspimport` utility.

Also, the `keytst` and `csptest` utilities have gained an extra command-line parameter. `keytst --help` now gives output containing the following information:

```
Key creation flags (only valid with -cx or -cs):
-e, --export               Create the key(s) with the 'exportable' bit set.
  -L, --length=BITLEN      Specify the new key length (default = 1024).
  -C, --counter            Create key counters (if supported).
  -K, --kitb               Create NVRAM-stored key(s) (if supported).
```

The `-C` and `-K` options require you to insert your ACS.

The command `csptest --help` outputs the following usage message:

```
Program options:
  -f, --flood               Run a continuous signature test.
  -d, --dsa                 Use DSA signatures rather than RSA signatures.
  -m, --ms                  Use the MS AES provider rather than nCipher's one
                             (possibly with modexp offload).
  -C, --counters            Generate keys with counters (needs NVRAM and ACS).
  -K, --kitb                Generate keys using KITB (needs NVRAM and ACS).
```

The `csputils` utility displays the NVRAM status of keys using the `--detail` option.

11. Container Storage Format

Versions of the CSP later than 1.11.0 have an updated container storage mechanism. CSP containers are now stored as part of the Security World instead of in the Windows registry file.

Versions of the CSP later than 1.11.0 use a non-backwards-compatible container and key storage format. If you are installing version 1.11.0 or later of the CSP over older versions, you must run the `cspmigrate` utility in order to convert containers and keys from the old system to the new system.

CSP versions 1.11.0 and later have a number of advantages over older versions:

- The CSP state is easily mirrored between multiple machines simply by copying the con tents of the Key Management Data directory or by sharing the Key Management Data directory across a network.
- The CSP key files can have arbitrary names (previously, the names of key files were linked to their key type and their container name). This new method facilitates the importation of existing Security World keys into the CSP.
- Every different container is now guaranteed to have a distinct storage location. There were circumstances in CSP versions older than 1.11.0 in which two containers with simi lar names could have shared the same keys wrongly.

However, there are some points to bear in mind concerning CSP versions 1.11.10 and later:

- If you want to share the same key between multiple computers, we supply the `cspim-port` utility for transferring keys between containers.
- Any existing containers with older versions of the CSP must be migrated to the new for mat. We provide a utility, `cspmigrate`, to migrate containers from the old to the new sys tem.