Bring Your Own Key for Microsoft Azure Key Vault

nShield® HSM Integration Guide

2023-12-05

Table of Contents

Chapter 1. Introduction

This *Integration Guide* is part of the Bring Your Own Key (BYOK) Deployment Service Package for Microsoft Azure. It covers the creation and transfer of a cryptographic key for use with Azure Key Vault.

This cryptographic key is known as a tenant key if used with the Azure Rights Management Service and Azure Information Protection. The key is created within the protection of the nShield Security World on the customer’s premises. It is then securely transferred to Microsoft Azure and the protection of an nShield Security World which is hosted within Azure via Key Vault.

This guide assumes that there is no existing Security World to create the key in. It contains steps to create a Security World using the supplied Entrust nShield Edge Hardware Security Module (HSM).

If a Security World already exists, parts of this guide can still be used for the generation and subsequent transfer of the tenant key. The benefits of using an nShield Hardware Security Module (HSM) with the Azure Key Vault include:

- Secure storage of the private key.
- FIPS 140 Level 3 validated hardware.

1.1. Product configurations

Entrust has successfully tested the use of an nShield HSM to generate and transfer a key into a Microsoft Azure Key Vault in the following configurations:

Product	Version
Microsoft OS	Windows 10
BYOK Preparation Tools	1.2
Security World software	12.80.4
nShield Compatibility Package	1.1.0
nShield support	Edge

If migrating from a tenant key managed by Microsoft to BYOK and you are using Microsoft Office 2010, you will need to contact

> Microsoft Support before proceeding with BYOK. This is because Microsoft Office 2010 with Azure RMS requires some additional configuration steps prior to migration to BYOK. Microsoft Support can be contacted here: https://docs.microsoft.com/en-us/rights-management/get-started/information-support#tocontact-microsoft-support.

1.2. Requirements

1.2.1. Before starting the integration process

Familiarize yourself with:

- The documentation for the nShield HSM.
- The documentation and setup process for the Azure Key Vault:
  - Planning and Implementing your Azure Rights Management Tenant Key, see http://technet.microsoft.com/en-us/library/dn440580.aspx.
  - Operations for your Azure Rights Management Tenant Key, see https://technet.microsoft.com/en-us/library/dn592126.aspx.

1.2.2. Before using nShield hardware and software

The following preparations are required before using nShield products:

- Obtain enough blank smartcards to create the Administrator Card Set (ACS). Six cards are delivered with the nShield Edge HSM.
- Define the Security World parameters as part of the preparation stage of the BYOK installation. For details of the security implications of the choices, see the *nShield Security Manual*.

Setting	Description
FIPS 140 Level	Sets the operational compliance level of the HSM. Certain operations (such as key generation) require any card from the Security World to be presented prior to the operation taking place. It also enables only the FIPS-approved mechanisms. It should be noted that this setting does not improve security. Level 3 should be chosen only if there is a regulatory requirement to do so.
ACS quorum size (K-of-N)	Specifies the number of cards in the ACS (N) and the number of cards required to instantiate the Security World (the quorum or K). It is good practice to choose a K number that is slightly greater than 50% of the value of N and have the N number provide a degree of resiliency in the unlikely event of card failure. For example, 2 of 3, 3 of 5, 4 of 7.
Cipher suite	Sets the symmetric algorithm to be used for the Security World module key. The choices are AES or AES (SP800-131A compliant).
Delegation	Sets the required quorum of cards from the ACS for various operation such as setting the real time clock (RTC) and allowing read/write access to NVRAM. The default is to use the same quorum (K) value as that needed to instantiate the Security World.
Key recovery	Determines whether application keys can be recovered if the Softcard protecting the application key is lost. This is on by default.
Passphrase recovery	Determines whether passphrases in use with Softcards can be replaced without knowing the original passphrase. This is off by default. Turning this on requires a quorum of the ACS to authorize the passphrase change without knowing the original passphrase.

- For creation of the Security World, determine who within the organization will act as custodians of the ACS cards and their attendance at the key generation ceremony.

1.2.3. Before using Microsoft Azure Key Vault with nShield software

The following preparations need to be made before starting to use Microsoft Azure with nShield software:

- Set up a standalone computing device. For example, a laptop with a suitable operating system installed and updated with all available security patches. This will be used for initial generation of the key. This device will not be connected to the network.
- Set up a second different computing device. For example, another laptop, with a suitable operating system installed and updated with all available security patches, and which is connected to the internet. This will be used for transferring the key to Microsoft Azure.
- Obtain access, including the passwords or passwords, to the true local administrator account (SID: S-1-5-21-<local identifier>-500) on both computing devices.
- Install the following software on the second different computing device:
  - Microsoft .NET Framework v4.7.2 Runtime (or later), see https://dotnet.microsoft.com/download. .NET is pre-installed on Windows 10
  - Windows Management Framework 5.1, which includes Windows PowerShell, see https://www.microsoft.com/en-us/download/details.aspx?id=54616. PowerShell is preinstalled on Windows 10.
- Ensure that you can download the following software packages. They will be needed during BYOK installation on the Internet-connected laptop:
  - Azure PowerShell (`Az` module) via `PowerShellGet`, see https://docs.microsoft.com/en-us/powershell/azure/install-az-ps.
  - `AIPService` module (if using Azure Key Vault with Azure Rights Management/Information Protection), see https://docs.microsoft.com/en-us/azure/information-protection/install-powershell.
- Obtain a Microsoft ID or Windows Live ID.
- Obtain an Azure account. If Azure RMS is to be used, this Azure account must support RMS.

- Obtain access to an activated Azure Key Vault that supports HSMs, see https://azure.microsoft.com/en-gb/pricing/details/key-vault/.
- Ensure that the Microsoft Azure AD has the Premium SKU License installed. This is required to use HSM backed keys with Azure Key Vault.
- Ensure that Azure Information Protection is set up within the Azure instance. This is required before the permissions for the Azure IP Service Principal can be added.
- Obtain a portable USB storage device or similar. It is recommended that a completely new storage device is used to ensure it is free from malware.
- Ensure that you can download the BYOK Preparation Tools for Microsoft Key Vault, see https://www.microsoft.com/en-us/download/details.aspx?id=45345. Ensure that the set of tools are appropriate for the Azure instance or region to be used.
- Ensure that you can download the Microsoft Visual C++ 2013 (12.0.30501) Redistributable Package. This may or may not be required.
  - 64 bit version: https://download.microsoft.com/download/2/E/6/2E61CFA4-993B-4DD4-91DA-3737CD5CD6E3/vcredist_x64.exe.
  - 32-bit version: https://download.microsoft.com/download/2/E/6/2E61CFA4-993B-4DD4-91DA-3737CD5CD6E3/vcredist_x86.exe.

> Entrust recommends that you allow only unprivileged connections unless you are performing administrative tasks.

1.2.4. Considerations for keys

- The key name must be lower-case.
- Only 1024-bit or 2048-bit RSA keys are supported.

> 1024-bit keys are not recommended and should only be used if migrating an existing on-premises RMS key protected by a nShield HSM to Azure RMS.

1.3. Supported nShield hardware and software versions

Entrust has successfully tested with the following nShield hardware and software versions:

1.3.1. Edge

Security World software	Firmware
12.80.4	12.50.8
12.80.4	12.72.0
12.80.4	12.60.6
12.71.0	12.50.8
12.71.0	12.60.6

1.4. Supported nShield functionality

Module key protection or Softcards are used. OCS with BYOK is not supported.

Feature	Support
Key Generation	Yes
Key Management	Yes
Key Import	Yes
Key Recovery	Yes
1-of-N Operator Card Set	No
K-of-N Operator Card Set	No
Softcards	Yes
Module-only Key	Yes
FIPS 140 Level 3 Support	Yes
Load Sharing	Yes
Fail Over	Yes

1.5. More information

For more information about OS support, contact your Azure sales representative or Entrust nShield Support, https://nshieldsupport.entrust.com.

> Access to the Entrust nShield Support Portal is available to customers under maintenance. To request an account, contact nshield.support@entrust.com.

Chapter 2. Procedures

Integration procedures include:

- Set up of Security World and the nShield Edge HSM locally.
- Generate the tenant key within the confines and protection of the local Security World.
- Use the BYOK Preparation Toolset to modify the permissions assigned to the tenant key such that it can be securely transferred to Microsoft. The tenant key never leaves the confines of the HSM in plaintext.
- Transfer of tenant key to Microsoft Azure.
- Perform attestation that:
  - The key transfer has been successful.
  - The tenant key has appropriate permission such that it cannot be extracted from the Microsoft Azure based Security World, nor used by Microsoft, and that it is hosted within verifiable nShield HSMs.

This integration will be conducted on two separate computing devices:

- An online computing device which will manage the Azure account.
- An offline computing device which will be used to generate and store the tenant key.

2.1. Prepare the internet-connected computing device

This section describes preparation of the internet-connected computing device and getting ready to transfer the generated tenant key to Microsoft Azure.

> ℹ On Windows machines, ensure that the minimum required version of both the .NET Framework and Windows Management Framework are installed prior to continuing with this guide.

> ℹ These instructions assume Windows 10 is being used. They may require modification if Windows 8.1 or another supported operating system is being used instead.

2.1.1. Install the Azure PowerShell module

1. Launch a PowerShell console with administrator privileges. Right-click the Windows icon in the lower left corner of the screen, and select **Windows**

PowerShell (Admin).

2. Check the version of PowerShell. It must be at least 5.1.

   ```
   $PSVersionTable.PSVersion

   Major  Minor  Build  Revision
   -----  -----  -----  --------
   5      1      17134  765
   ```

3. Enable the running of scripts, which is disabled by default. Enter **A** for yes to all.

   ```
   Set-ExecutionPolicy RemoteSigned
   ```

4. The `Set-ExecutionPolicy` command allows scripts to run but requires a digital signature on all scripts downloaded from the internet. All Microsoft scripts are digitally signed. Local scripts can be run even if they are not digitally signed. Check that the digital signature requirement has been set correctly:

   ```
   Get-ExecutionPolicy
   ```

5. Install `Az`, a rollup module for the Azure PowerShell cmdlets:

   ```
   Install-Module -Name Az -AllowClobber
   ```

6. A message should appear stating that the NuGet provider is required. If NuGet has already been installed, this message may not be seen.

   If the message appeared, type **Y** to allow the NuGet provider installation to continue.

7. If this is the first `PowerShellGet` module to be installed, a message will be shown about an Untrusted Repository. This is expected. Do one of the following:
   - Type **A** (for Yes to All), and select **Enter** on the keyboard.
   - Mark the PowerShell Gallery as trusted:
     a. Type **N** at the prompt
     b. Enter the following command:

        ```
        Set-PSRepository -Name PSGallery -InstallationPolicy Trusted
        ```

     c. If the NuGet installations fails, try the following commands:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12 Install-
PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force
```

   d. Run the `Install-Module -Name Az -AllowClobber` command again.

8. During the installation, you should see a number of packages installed:

```
Installing package 'Az'
   Installing dependent package 'Az.Cdn'
   [ooooooooooooooooo                                       ]
  Installing package 'Az.Cdn'
     Process Package Manifest
     [oooooooooooooooooooooooooooooooooooooooooooooooo      ]
```

   When the `Az` module and other dependencies installed, the PowerShell command prompt returns.

9. The `Az` module needs to be imported before it can be used:

```
Import-Module Az
```

   Importing may take some time. The command prompt will return when the module has completed loading.

10. Check that the `Az` module is properly loaded and ready for use:

```
Get-Module -Name Az*
```

   The output should be similar to the following:

```
ModuleType Version    Name                      Exported Commands
---------- -------    ----                      -----------------
Script     2.1.0      Az
Script     1.5.2      Az.Accounts               {Add-AzEnvironment, Clear-AzContext...
Script     1.0.1      Az.Aks                    {Get-AzAks, Import-AzAksCredential,...
Script     1.1.0      Az.AnalysisServices       {Add-AzAnalysisServicesAccount, Exp...
Script     1.1.0      Az.ApiManement            {Add-AzApiManagementApiToProduct, A...
Script     1.0.0      Az.ApplicationInsights    {Get-AzApplicationInsights, Get-AzA...
Script     1.2.2      Az.Automation             {Export-AzAutomationDscConfiguratio...
Script     1.1.0      Az.Batch                  {Disable-AzBatchAutoScale, Disable-...
Script     1.0.0      Az.Billing                {Get-AzBillingInvoice, Get-AzBillin...
Script     1.2.0      Az.Cdn                    {Confirm-AzCdnEndpointProbeURL, Dis...
Script     1.1.1      Az.CognitiveServices      {Get-AzCognitiveServicesAccount, Ge...
Script     2.1.0      Az.Compute                {Add-AzContainerServicesAgentPoolPr...
Script     1.0.1      Az.ContainerInstance      {Get-AzContainerGroup, Get-AzContai...
Script     1.0.1      Az.ContainerRegistry      {Get-AzContainerRegistry, Get-AzCon...
Script     1.1.1      Az.DataFactory            {Get-AzDataFactory, Get-AzDataFacto...
Script     1.0.0      Az.DataLakeAnalytics      {Add-AzDataLakeAnalyticsDataSource,...
```

11. Leave the PowerShell console open ready for use.

2.1.2. Install the Azure Information Protection Service module

To install the Azure Information Protection Service (AIPService) module (if required):

1. In the PowerShell console, run the following command:

   ```
   Install-Module -Name AIPService
   ```

2. If the PowerShell Gallery was not marked as a trusted repository, then you will be prompted to approve the install from an untrusted repository.

   Type **A** at the prompt, and select **Enter** on the keyboard.

3. Load the module:

   ```
   Import-Module AIPService
   ```

4. Check that the module is correctly loaded:

   ```
   Get-Module AIPService
   ```

   The output should be similar to the following:

   ```
   ModuleType Version   Name
   ---------- -------   ----
   Binary     1.0.0.1   APIService
   ```

2.1.3. Get the Azure Active Directory Tenant ID

1. From the command prompt, sign in to your Azure account:

   ```
   Connect-AzAccount
   ```

2. In the dialog that appears, enter your Azure username and password.

   If you are prompted whether you want to **Run software from this untrusted publisher**, press **A** for "Always Run".

   The initial connection may take some time as the subscription information is retrieved from Azure.

   You should see the Azure account, subscription name, and other information displayed when successfully connected.

3. The Tenant ID is the Subscription ID used in subsequent steps.

   If only a single subscription is listed, then this will be the subscription to use with BYOK.

   If multiple subscriptions are associated with the account, use the following command to display them all:

   ```
   Get-AzSubscription
   ```

   The output should be similar to the following:

   ```
   Name                           Id                        TenantId
   ----                           --                        --------
   Visual Studio Enterprise - MPN 2b3c9783-xxxxxxxx...   5bd48eab-xxxxxxxx...
   Visual Studio Enterprise - MPN 01f55689-xxxxxxxx...   5bd48eab-xxxxxxxx...
   ```

   Make a note of the Subscription ID in the ID column. It will be used with Azure Key Vault as it is required later.

4. Leave the Azure session open as it will be used later to upload the Key Vault key.

2.1.4. Download the BYOK Package for the relevant Azure region

There are eighteen BYOK packages:

- `KeyVault-BYOK-Tools-AsiaPacific.zip`
- `KeyVault-BYOK-Tools-Australia.zip`
- `KeyVault-BYOK-Tools-Canada.zip`
- `KeyVault-BYOK-Tools-Europe.zip`
- `KeyVault-BYOK-Tools-France.zip`
- `KeyVault-BYOK-Tools-Germany.zip`
- `KeyVault-BYOK-Tools-Germany-Public.zip`
- `KeyVault-BYOK-Tools-India.zip`
- `KeyVault-BYOK-Tools-Japan.zip`
- `KeyVault-BYOK-Tools-Korea.zip`
- `KeyVault-BYOK-Tools-LatinAmerica.zip`
- `KeyVault-BYOK-Tools-SouthAfrica.zip`
- `KeyVault-BYOK-Tools-Switzerland.zip`

- `KeyVault-BYOK-Tools-UAE.zip`
- `KeyVault-BYOK-Tools-UnitedKingdom.zip`
- `KeyVault-BYOK-Tools-UnitedStates.zip`
- `KeyVault-BYOK-Tools-USGovernment.zip`
- `KeyVault-BYOK-Tools-USGovernmentDoD.zip`

Each package contains a number of files:

- A Key Exchange Key (KEK) package.
- A Security World package.
- A Python script for verifying keys (key attestation).
- A command line executable and supporting DLLs.
- A Visual C++ redistributable package.

To download the BYOK Package:

1. Navigate to https://www.microsoft.com/en-us/download/details.aspx?id=45345 and download the correct package for the region in which Azure Key Vault is to be used.
2. Verify the integrity of the downloaded BYOK Preparation Toolset with the `Get-FileHash` cmdlet:

   ```
   Get-FileHash KeyVault-BYOK-Tools-region.zip
   ```

   region is the location that the BYOK Preparation Toolset.

   ```
   Get-FileHash KeyVault-BYOK-Tools-Europe.zip
   ```

   Compare the hash given in the output from the `Get-FileHash` command to the one given for the BYOK package below and ensure they are the same.

3. The hashes for each package are as follows (SHA 2):

   ```
   4BC14059BF0FEC562CA927AF621DF665328F8A13616F44C977388EC7121EF6B5KeyVault-BYOK-Tools-AsiaPacific.zip
   CD0FB7365053DEF8C35116D7C92D203C64A3D3EE2452A025223EEB166901C40AKeyVault-BYOK-Tools-Australia.zip
   61BE1A1F80AC79912A42DEBBCC42CF87C88C2CE249E271934630885799717C7BKeyVault-BYOK-Tools-Canada.zip
   9AAA63E2E7F20CF9BB62485868754203721D2F88D300910634A32DFA1FB19E4AKeyVault-BYOK-Tools-Europe.zip
   5C9D1F3E4125B0C09E9F60897C9AE3A8B4CB0E7D13A14F3EDBD280128F8FE7DFKeyVault-BYOK-Tools-France.zip
   5385E615880AAFC02AFD9841F7BADD025D7EE819894AA29ED3C71C3F844C45D6KeyVault-BYOK-Tools-Germany.zip
   54534936D0A0C99C8117DB724C34A5E50FD204CFCBD75C78972B785865364A29KeyVault-BYOK-Tools-Germany-Public.zip
   49EDCEB3091CF1DF7B156D5B495A4ADE1CFBA77641134F61B0E0940121C436C8KeyVault-BYOK-Tools-India.zip
   3933C13CC6DC06651295ADC482B027AF923A76F1F6BF98B4D4B8E94632DEC7DFKeyVault-BYOK-Tools-Japan.zip
   71AB6BCFE06950097C8C18D532A9184BEF52A74BB944B8610DDDA05344ED136FKeyVault-BYOK-Tools-Korea.zip
   E7DFAFF579AFE1B9732C30D6FD80C4D03756642F25A538922DD1B01A4FACB619KeyVault-BYOK-Tools-LatinAmerica.zip
   C41060C5C0170AAAAD896DA732E31433D14CB9FC83AC3C67766F46D98620784AKeyVault-BYOK-Tools-SouthAfrica.zip
   88CF8D39899E26D456D4E0BC57E5C94913ABF1D73A89013FCE3BBD9599AD2FE9KeyVault-BYOK-Tools-Switzerland.zip
   ```

```
FADE80210B06962AA0913EA411DAB977929248C65F365FD953BB9F241D5FC0D3KeyVault-BYOK-Tools-UAE.zip
432746BD0D3176B708672CCFF19D6144FCAA9E5EB29BB056489D3782B3B80849KeyVault-BYOK-Tools-UnitedKingdom.zip
2E8C00320400430106366A4E8C67B79015524E4EC24A2D3A6DC513CA1823B0D4KeyVault-BYOK-Tools-UnitedStates.zip
A79DD8C6DFFF1B00B91D1812280207A205442B3DDF861B79B8B991BB55C35263KeyVault-BYOK-Tools-USGovernmentDOD.zip
F8DB2FC914A7360650922391D9AA79FF030FD3048B5795EC83ADC59DB018621AKeyVault-BYOK-Tools-USGovernment.zip
```

These hashes can also be viewed online at https://docs.microsoft.com/en-us/azure/key-vault/keys/hsm-protected-keys.

4. Copy the downloaded package to the USB storage device or similar.

2.2. Install the nShield software and configure the HSM

This section describes installation of the nShield software, attaching the HSM to the standalone computing device, generating the Security World container and associated Administrator Card Set (ACS) for protection of the tenant key, configuration of all parameters associated with the Security World and its card sets and integration of the HSM with the Cryptographic Service Provider (CSP).

All procedures in this section should be completed on the standalone computing device.

2.2.1. nShield software installation

1. Make sure you have Administrator privileges. This is needed to install software, including the nShield packages.
2. Insert the nShield software media into the DVD drive on the standalone computing device, or mount the ISO.
3. Open a file explorer window, and browse to the root directory of the media or ISO.
4. Select `Setup.msi` and run it.
5. Ensure that the following minimum options are selected to be installed on the local hard drive:
   - nShield Hardware Support
   - nShield Core Tools
6. Select **Install**.
7. The software will install, and then a set of further confirmation dialogs may be presented. If any are presented, accept all the default parameters for these.
8. Select **Finish** to close the installation wizard.

9. Unzip Compatibility_12.40_Win.
10. Move the files from the unzipped folders to the relevant folders under C:\Program Files\nCipher\nfast.
11. Check that C:\program Files\nCipher\nfast\bin\new-world-1240.exe exists.

2.2.2. Attach the HSM

1. Take the HSM out of its box. Check the tamper evident hologram in the lower right corner of the HSM front panel.

   If this appears scratched, damaged, or is not present, do not use the HSM and contact Entrust nShield Support immediately, see https://nshieldsupport.entrust.com.

   > Access to the Entrust nShield Support Portal is available to customers under maintenance. To request an account, contact nshield.support@entrust.com.

2. Remove all other protective packaging from the HSM.
3. Take a note of the Paper Serial Number (PSN). It can be found on the back of the Edge beneath the pull out stand and is also written on the side of the HSM packaging box. It usually begins with 06.

   ```
   PSN:
   ```

4. Store this PSN in a place that is accessible by all engineering staff, because they may need this number if they need to obtain support for the HSM.
5. Attach the USB Standard-B connector into the HSM.

   A compatible USB cable is found inside the packaging for the HSM.

6. Attach the USB Standard-A connector to the target host where the support software has been previously installed.

   If this is the first time the device has been connected, you may see USB drivers installed. Wait until the driver installation has completed.

7. It is recommended that you add the working directories for the nShield Security World binaries and Python binaries to the PATH environment variable. This negates the need to keep changing to the correct working directory to use either the Security World binaries or Security World provided Python binaries, which are used with the BYOK Preparation Toolset.

   a. Right-click the Windows icon and choose **System** from the menu. In the **Settings** window, scroll down to the **Related Settings** section and select the link for **System info**.
   b. In the **System** window that appears, select **Advanced System Settings**, and then in the **System Properties** window, select **Environment Variables**.
   c. In the **Environment Variables** window, scroll down in the **System variables** section, and double-click the **Path** variable.
   d. In the **Edit environment variable** window, select **New**, and enter:

   ```
   %NFAST_HOME%\bin
   ```

   e. Select **New** again, and enter:

   ```
   %NFAST_HOME%\python
   ```

   f. Select **OK** when both entries have been added and close all open windows.

8. Open a command prompt with local administrator privileges:
   a. In the **Start** menu, type `cmd`.
   b. Right-click the **Command Prompt** app, and select **Run as Administrator**.
9. Run the `enquiry` command. Check that there is output from both the **server** and **module #1**.
10. If **module #1** fails or there is no output shown from **module #1**.
   a. Restart the hardserver using one of the following methods:
      - To use the Windows Services GUI, At the command prompt, enter:

        ```
        run services.msc
        ```

        Find the `nFast Server` service, stop it, then restart it.

      - From the command line:

        ```
        net stop "nfast Server" && net start "nfast Server"
        ```

   b. Run the `enquiry` command and check whether **module #1** is in the operational state.
11. When **module #1** is shown as **operational**, check its firmware version. If firmware updates are necessary, perform them now. For instructions, see the *nShield Edge User Guide*.

2.2.3. Create the Security World

1. Ensure that the Security World parameters have been defined and the ACS custodians are present.
2. If it is not open yet, open a command prompt with local administrator privileges.
    a. In the **Start** menu, type `cmd`.
    b. Right-click the **Command Prompt** app, and select **Run as Administrator**.
3. Put the HSM into initialization mode:
    a. On the HSM, press the **Status** button until the orange **I** light is blinking. The orange **I** light should be solid, not blinking, and the blue **Clear** light should be blinking.
    b. Press the **Clear** button.
4. Create a standard AES module key with key recovery and passphrase recovery options enabled. Run the following command, replacing K/N with your preferred ACS quorum size:

    ```
    new-world-1240 --initialize --module=1 --cipher-suite=<cipher suite> --acs-quorum=K/N p
    ```

5. You will now be prompted to insert N blank/new/formatted smartcards. In turn, have your ACS custodians present their allocated card, noting the serial number of the smartcard and the corresponding passphrase.

    Each card and passphrase should be stored in separate tamper-resistant envelopes and should be dated and signed.

    The cards and passphrases should not be sealed until the end of the tenant key ceremony, in case they are needed later during the ceremony.

6. Once the command exits, reset the HSM to operational mode:
    a. Press the **Status** button on the HSM until the green **O** light is blinking.

       The green **O** light should be solid, not blinking, and the blue **Clear** light should be blinking slowly.

    b. Press the **Clear** button.
7. Run the `nfkminfo` command to view details about the newly created Security World.
8. Check that the module state is **usable**.

If you are using Remote Administrative Cards, you may need to

> generate a remote administration warrant. Contact nShield support for assistance.

2.2.4. Install the BYOK package

1. Ensure that the BYOK package is available for copying onto the USB storage device.
2. Connect the USB storage device to the standalone computing device.
3. Copy the BYOK package onto the USB device, to a suitable location.
4. Extract the contents of the package to a suitable location.
5. Check **Add/Remove Programs** to see if the Visual C++ 2013 (x86 and x64) runtime components are installed. If these are already installed, skip the rest of this step. If they are not installed, double-click `vcredist_x64.exe` from the extract location and follow the instructions to complete the wizard.

2.2.5. Verify the BYOK package

Verification ensures that:

- The Key Exchange Key has been generated on a genuine nShield HSM.
- The Azure Key Vault Security World has been generated on a genuine nShield HSM.
- The Key Exchange Key is defined as non-exportable.
    1. Change to the directory to which the BYOK package was extracted to.
    2. Type the correct verification command for the package downloaded:

    ```
    python verifykeypackage.py -k BYOK-KEK-pkg-REGIONNAME-1 -w BYOK-SecurityWorld-pkg-REGIONNAME-1 (North
    America)
    ```

    Where *REGIONNAME* is one of the following:

REGIONNAME	**Region**
NA	North America
EU	Europe
AP	Asia Pacific
LATAM	Latin America

REGIONNAME	**Region**
JPN	Japan
AUS	Australia
USGOV	Azure US Government
CANADA	Canada
GERMANY	Germany
INDIA	India
KOREA	Korea
SA	South Africa
UAE	United Arab Emirates
FRANCE	France
UK	United Kingdom
USDOD	US Government Department of Defense

3. If running `verifykeypackage.py` results in the following error:

   ```
   "msvcr120.dll is missing…"
   ```

   Then download the Microsoft Visual C++ 2013 (12.0.30501) Redistributable Package, and install it.

- 64-bit version: https://download.microsoft.com/download/2/E/6/2E61CFA4-993B-4DD4-91DA-3737CD5CD6E3/vcredist_x64.exe.
- 32-bit version: https://download.microsoft.com/download/2/E/6/2E61CFA4-993B-4DD4-91DA-3737CD5CD6E3/vcredist_x86.exe.

When the C++ package is installed, run the `verifykeypackage.py` command again.

1. Confirm that **RESULT: SUCCESS** is shown in the command window
2. Optionally, you can look further up the output of this command to see the verification steps undertaken on the Key Encryption Key (KEK) from the Microsoft regional Security World. The script verifies the key chain up to

the nShield HSM Root key hash. This hash is:

```
59178a47 de508c3f 291277ee 184f46c4 f1d9c639
```

This hash can be verified at https://www.entrust.com/digital-security/hsm/key-management/cloud-microsoftazure/validation.

2.2.6. Generate the tenant key

1. Generate the key:

```
generatekey --generate simple type=RSA size=2048 protect=module ident=KEYNAME plainname=KEYNAME nvram=no
pubexp=
```

KEYNAME in both locations stands for a suitable name, for your tenant key, for example `azuremstenantkey`.

No value is specified after the `pubexp` parameter. This is correct because the default value is used.

The command will create an encrypted key blob in `%NFAST_KMDATA%\local` with a filename starting `key_simple_KEYNAME`.

2.2.7. Back up the Security World and the tenant key

1. Open the `C:\ProgramData\nCipher\` directory:
2. Copy the entire `Key Management Data` folder to a safe backup location and ensure that the data is carefully stored.

   This data consists of the entire Security World data structure, Security World configuration, and information on configured Cardsets. This backup also contains the tenant key that was created in the earlier steps.

   This data is encrypted and cannot be used without a Security World software installation, HSM, and a quorum of the ACS and any associated passphrases.

2.3. Prepare the tenant key for secure transfer to Microsoft Key Vault

All procedures in this section should be completed on the standalone computing device, unless stated otherwise.

2.3.1. Reduce permissions on a copy of the tenant key

1. On the standalone computing device, change to the directory to where the BYOK package was extracted.
2. Run the `KeyTransferRemote` command in a command prompt window as administrator:

   ```
   KeyTransferRemote.exe -ModifyAcls -KeyAppName simple -KeyIdentifier KEYNAME -ExchangeKeyPackage BYOK-KEK-
   pkg-REGIONNAME-1 -NewSecurityWorldPackage BYOK-SecurityWorld-pkg-REGIONNAME-1
   ```

   KEYNAME is the name of the tenant key, for example `azuremstenantkey`.

   REGIONNAME is appropriate to the regional instance of Azure, see the table in Verify the BYOK package.

   You will be asked to provide a quorum from the ACS and any appropriate passphrases.

3. If you receive the following error when running the `KeyTransferRemote` commands:

   ```
   nCipher command returned error code: TimeLimitExceeded
   ```

   Remove any member of the ACS that is in the card slot in the Edge HSM and run the `KeyTransferRemote` command again. Then, reinsert the first card of the ACS quorum when prompted.

4. If the `KeyTransferRemote` command reports a failure, check the Application event log.

   If one of the following errors are reported:

   ```
   Activation context generation failed for "C:\Users\Administrator\Downloads\AzureRMS-BYOK-Tools-
   REGIONNAME\AzureRMS-BYOK-Tools-REGIONNAME\nCipherClr.dll
   Could not load file or assembly nCipherClr.dll
   ```

   Download the Microsoft Visual C++ 2013 (12.0.30501) Redistributable Package, and install it.

   - 64-bit version: https://download.microsoft.com/download/2/E/6/2E61CFA4-993B-4DD4-91DA-3737CD5CD6E3/vcredist_x64.exe.
   - 32-bit version: https://download.microsoft.com/download/2/E/6/2E61CFA4-993B-4DD4-91DA-3737CD5CD6E3/vcredist_x86.exe.

   When the C++ package is installed, run the `KeyTransferRemote` command again.

5. When the KeyTransferRemote command completes, ensure that **RESULT: SUCCESS** is displayed.

   The modified tenant key will have the name: key_xferacId_KEYNAME.

   This will be stored in the C:\ProgramData\nCipher\Key Management Data\local directory.

   A log file is also produced and stored in the directory to which the BYOK package was extracted. This file is called ModifyAcls-key_xferacld_KEYNAME.

2.3.2. Confirm the reduced permissions on the tenant key

Entrust provides utility programs for proving that the tenant key has been modified with reduced permissions. The much reduced permissions restrict what Microsoft can do with the customer tenant key. For example, Microsoft cannot export the key in plain text, copy the key or move the key to a different Security World, for example in a different Azure region.

The output of the utility programs can be run and the output of each checked. Alternatively, the output can be piped to a text file and stored for later verification or audit purposes.

1. Run the following commands:

```
"%NFAST_HOME%\bin\preload.exe" --module=1 --appname=simple --key-ident=KEYNAME
"%NFAST_HOME%\python\python"
"%NFAST_HOME%\python\examples\aclprint.py"

"%NFAST_HOME%\bin\preload.exe" --module=1 --appname=xferacld --key-ident=KEYNAME
"%NFAST_HOME%\python\python"
"%NFAST_HOME%\python\examples\aclprint.py"
```

   If any of the commands returns an error, contact Entrust nShield Support, https://nshieldsupport.entrust.com.

   > Access to the Entrust nShield Support Portal is available to customers under maintenance. To request an account, contact nshield.support@entrust.com.

   KEYNAME_KEYNAME is the name of the tenant key, for example azuremstenantkey.

   If you want to pipe the output of these commands elsewhere, add the piping command to the end of the commands above:

```
"%NFAST_HOME%\bin\preload.exe" --module=1 --appname=simple --key-ident=azuremstenantkey
"%NFAST_HOME%\python\python.exe
"%NFAST_HOME%\python\examples\aclprint.py" ^
>> "C:\ProgramData\nCipher\Key Management Data\local\key_simple_azuremstenantkey.txt"

"%NFAST_HOME%\bin\preload.exe" --module=1 --appname=xferacld --key-ident=azuremstenantkey
"%NFAST_HOME%\python\python.exe"
"%NFAST_HOME%\python\examples\aclprint.py" ^
>> "C:\ProgramData\nCipher\Key Management Data\local\key_xferacld_azuremstenantkey.txt"
```

> If the output includes an error with these commands, contact nShield support.

2. Compare the **OpPermissions** lines in the files created in the previous step. Note that only a single group of permissions now exists: `acl.groups[0]` only in the xferacld version of the key as opposed to `acl.groups[0]` and `acl.groups[1]` for the original key. The remaining are much decreased in the modified `xferacld` version of the key that will be used as the tenant key.

   The following error message can be ignored:

   ```
   > preload: error: option --appname: invalid choice: 'xferacld' (choose from 'custom', 'embed', 'hwcrhk',
   'kpm', \
   'pkcs11', 'seeconf', 'seeinteg', 'seessl', 'simple')
   ```

2.3.3. Encrypt your tenant key with Microsoft's Key Encryption Key

1. Open a command prompt window as an administrator.
2. Change to the directory into which you extracted the BYOK package.
3. Run the command appropriate to the regional instance of Azure RMS that you will be using and the BYOK package you previously downloaded.

   ```
   KeyTransferRemote.exe -Package -KeyIdentifier KEYNAME -ExchangeKeyPackage  BYOK-KEK-pkg-REGIONNAME-1
   -NewSecurityWorldPackage BYOK-SecurityWorld-pkg-REGIONNAME-1 -SubscriptionId GUID/SubscriptionID
   -KeyFriendlyName FILENAME
   ```

   KEYNAME is the name of the tenant key, for example `azuremstenantkey`.

   REGIONNAME is appropriate to the regional instance of Azure, see the table in Verify the BYOK package.

   GUID/SubscriptionID is the BPOSId or Subscription ID discovered in Get the Azure Active Directory Tenant ID.

   FILENAME is a name that describes the BYOK package to be transferred to

Microsoft, for example `azuremstenantkey`.

4. Check that **Result:SUCCESS** is displayed.

   The outputted BYOK package will be in the current working folder with a file name similar to `KeyTransferPackage-<FILENAME>.byok`.

5. Copy the BYOK package to the USB storage device and from the USB storage device to a suitable location on the internet-connected computing device.

2.4. Transfer the modified tenant key to Azure Key Vault

All procedures in this section should be completed on the internet-connected computing device.

2.4.1. Create a new Resource Group

1. On the internet-connected computing device, ensure that you are signed in to Azure.
2. Open a PowerShell session and sign in to your account:

   ```
   Connect-AzAccount
   ```

   Enter your Azure account username and password when prompted.

3. If multiple subscriptions are available, set the Subscription ID to use:

   ```
   Set-AzContext "Subscription ID"
   ```

   This must be the same Subscription ID that was used with the BYOK package.

4. Display the existing Resource Groups.

   ```
   Get-AzResourceGroup
   ```

5. Select an existing Resource Group or create a new one by running this command:

   ```
   New-AzResourceGroup -Name 'NAMEOFGROUP' -Location 'LOCATION'
   ```

NAMEOFGROUP is with the name of the Resource Group you want to create.

LOCATION is the Azure region where you want to create the group, for example `northeurope`.

To discover the list of valid locations, can use the `Get-Az-Location` command:

```
Get-AzLocation | Where-Object Providers -like "*keyvault*""
```

The location you use must match the location of the BYOK package you downloaded earlier.

2.4.2. Create a new Key Vault

If you already have a Key Vault that you want to use, you can skip this step.

1. Execute the `New-AzKeyVault` command:

```
New-AzKeyVault -VaultName 'NAMEOFVAULT' -ResourceGroupName 'NAMEOFGROUP' -Location 'LOCATION' -SKU
'Premium'
```

The output should be similar to this:

```
Vault Name :                        NAMEOFVAULT
Resource Group Name :               NAMEOFGROUP
Location :                          LOCATION
Resource ID :                       RESOURCE ID
Vault URI :                         VAULT URI
Tenant ID :                         TENANT ID
SKU :                               Premium
Enabled For Deployment? :           False
Enabled For Template Deployment? :  False
Enabled For Disk Encryption? :      False
Soft Delete Enabled? :
Access Policies :
Network Rule Set :
                                    Default Action : Allow
                                    Bypass : AzureServices
                                    IP Rules :
                                    Virtual Network Rules :
Tags :
```

Investigate any warnings before continuing.

- *NAMEOFVAULT* is the name you want to give the Key Vault. The Key Vault name used must be unique across Azure.
- *NAMEOFGROUP* is the same Resource Group name created in Create a new Resource Group.

- *LOCATION* should be the same Azure region as used in Create a new Resource Group.

Only certain subscriptions support keys for use with Key Vault protected by HSMs, see https://azure.microsoft.com/en-gb/pricing/free-trial/. To use Key Vault with an HSM, the SKU must be Premium.

2.4.3. Add Permission for Key Vault administrator to manage the Key Vault

1. Run the following command:

   ```
   Get-AzADUser
   ```

   Note down the `ObjectID`.

   You can also obtain the `ObjectID` in one of the following ways:

   - Sign in to the Azure Portal, and look for the value under the Azure administrator user account.
   - From the command line, execute:

     ```
     Get-AzADUser -SearchString "administrator user ID"
     ```

   - From the command line, execute:

     ```
     $(Get-AzADUser -Filter "UserPrincipalName").ObjectId
     ```

     The *UserPrincipalName* can be for example the UPN of administrator.

2. Add permissions to the Key Vault for the administrator so that they can upload the BYOK package. Use one of the following commands:

   ```
   Set-AzKeyVaultAccessPolicy -VaultName 'NAMEOFVAULT' -ObjectID 'GUID of Object to apply permissions to' -
   PermissionstoKeys get,list,update,create,import
   ```

   or

   ```
   Set-AzKeyVaultAccessPolicy -VaultName 'NAMEOFVAULT' -UserPrincipalName 'UPN of the Administrator account'
   -PermissionstoKeys get,list,update,create,import
   ```

3. Additional permissions will need to be added for any users, applications, or

services that will need to use the newly uploaded tenant key.

2.4.4. Upload the BYOK Package

1. Use the `Add-AzureKeyVaultKey` cmdlet to transfer the BYOK package to the Azure Key Vault:

    ```
    Add-AzKeyVaultKey -VaultName NAMEOFVAULT -Name KEYNAME -KeyFilePath 'PATHTOPACKAGEFILE\FILENAME.byok'
    -Destination 'HSM'
    ```

    FILENAME is the name of the key to be transferred to Key Vault.

    PATHTOPACKAGEFILE is the location of the BYOK package file.

    KEYNAME is the name of the key.

2. If the upload is successful, the output will show the uploaded key attributes, the permissions, and the URI that can be used by calling applications.
3. Note the URI for later use.

2.4.5. Enumerate Key Vault Keys

You can enumerate all available or previously used Azure Key Vault keys with the `Get-AzKeyVaultKey` command:

```
Get-AzKeyVaultKey -VaultName 'NAMEOFVAULT'
```

2.4.6. Start to use the transferred Key Vault Key

For introductory information about Azure Key Vault, see https://docs.microsoft.com/en-gb/azure/key-vault/general/overview.

For a full list of commands related to Azure Key Vault, see https://docs.microsoft.com/en-us/cli/azure/keyvault?view=azure-cli-latest.

2.5. Additional steps for Azure Information Protection

This is only required if you are setting up a tenant key for use by Azure IP.

> ℹ The commands listed in this section are for the legacy `AADRM`

> module. For more information, see https://docs.microsoft.com/en-us/powershell/module/aadrm/?view=azureipps.

2.5.1. Grant Azure RMS permissions to use the uploaded tenant key

1. Ensure that Azure Information Protection has been set up within the Azure instance.
2. Use the `Set-AzKeyVaultAccessPolicy` cmdlet to provide Azure IP with the necessary rights to use the tenant key uploaded to Key Vault:

```
Set-AzKeyVaultAccessPolicy -VaultName 'NAMEOFVAULT' -ResourceGroupName 'NAMEOFGROUP' -ServicePrincipalName
00000012-0000-0000-c000-000000000000 -PermissionsToKeys decrypt,encrypt,unwrapkey,wrapkey,verify,sign,get
```

2.5.2. Specify the URI to the tenant key

1. Ensure that the account you are going to use is able to use Azure AD Rights Management. It must also have global administrator account details for the Azure RMS tenant.
2. Sign in to Azure RMS/IP:

   With an Azure IP command:

   ```
   Connect-AipService
   ```

   With a legacy AADRM command:

   ```
   Connect-AadrmService
   ```

3. Provide your credentials when prompted.
4. Once successfully signed in, run the cmdlet.

   With an Azure IP command:

   ```
   Use-AipServiceKeyVaultKey -KeyVaultKeyUrl "https://NAMEOFVAULT.vault.azure.net:443/keys/NAMEOFKEY/VERSION"
   ```

   With a legacy ADDRM command:

   ```
   Use-AadrmKeyVaultKey -KeyVaultKeyUrl "https://<NAMEOFVAULT>.vault.azure.net:443/keys/NAMEOFKEY/VERSION"
   ```

You can run this cmdlet before or after the protection service (Azure Rights Management) is activated.

The URI in `-KeyVaultKeyUrl` can be copied from the output from Create a new Key Vault.

VERSION can be found by looking for the Version information provided from the output of the `Add-AzureKeyVaultKey` command run above. It looks like a random string of characters, for example `3e55fead11ae4a26bddb0171312d73c0`.

5. When your command successfully runs, the key is added as an archived customer-managed tenant key for Azure Information Protection for your organization. To make it the active tenant key for Azure Information Protection, you must run the `Set-AipServiceKeyProperties` or legacy `Set-AadrmKeyProperties` cmdlet, see Enable the Azure RMS service.

2.5.3. Enable the Azure RMS service

1. Ensure to be signed in as the global administrator. This is required for the AADRMS service.
2. Enable Azure RMS. This enables the protection service from Azure Information Protection so that all users in your tenant can protect documents and emails.:

   With an Azure IP command:

   ```
   Enable-AIPservice
   ```

   With a legacy AADRM command:

   ```
   Enable-Aadrm
   ```

   For more information, see https://docs.microsoft.com/en-gb/azure/information-protection/information-support#BKMK_SupportOptions.

   To enable different users for Rights Management, use the `Set-AadrmOnboardingControlPolicy` command, see https://docs.microsoft.com/en-us/rights-management/deploy-use/activate-service.

3. Check state of the new BYOK tenant key:

   ```
   Get-AipServiceKeys
   ```

An example output below shows two keys:

- The default Microsoft key.
- The new BYOK key that is still not Active.

```
KeyIdentifier :     2fdc841a-3191-4095-a992-534bfb54a5e6
CreationTime :      5/23/2018 9:04:56 AM
Status :            Active
KeyType :           Microsoft-managed
FriendlyName :      ABC Company
PublicKey :
KeyVaultKeyUrl :
KeyIdentifier :     231fa2a9-4178-48cb-9581-f3207feed308
CreationTime :      11/28/2019 2:31:51 PM
Status :            Archived
KeyType :           Customer-managed (BYOK)
FriendlyName :      2131fa2a9-4178-48cb-9581-f32307feed308
PublicKey :         {"n":"ymT-m07-···81AosuQ", "e":"AQAB"}
KeyVaultKeyUrl :    https:// NAMEOFVAULT.vault.azure.net:443/keys/
                    azuremstenantkey001/82a2a68ea03d4d54b57fb470b40f331f
```

4. Change state of the new key to Active:

   If the Azure Rights Management service is already activated, tell Azure Information Protection to use this key as the active tenant key for the Azure Rights Management service.

   If you do not do this step, Azure Information Protection will continue to use the default Microsoft-managed key that was automatically created for your tenant.

   ```
   Set-AipServiceKeyProperties -KeyIdentifier 'KEY-ID'-Active $true
   ```

   For example:

   ```
   Set-AipServiceKeyProperties -KeyIdentifier 231fa2a9-4178-48cb-9581-f3207feed308 -Active $true
   ```

   For more information, see https://docs.microsoft.com/en-us/azure/information-protection/plan-implement-tenant-key.

2.6. Start to use the transferred tenant key

1. Consider turning on Azure RMS logging, see https://docs.microsoft.com/en-us/azure/information-protection/log-analyze-usage. This will keep a record of all instances where the tenant key is used. This may be useful for auditing and/or troubleshooting purposes.

The following two request types found in logs indicate BYOK tenant key was used:

```
KeyVaultDecryptRequest
```

The client is attempting to decrypt the RMS-protected content. Applicable only for a customer-managed tenant key (BYOK) in Azure Key Vault.

```
KeyVaultSignDigest
```

A call is made when a customer-managed key (BYOK) in Azure Key Vault is used for signing purposes.

2. Analyze the state of configuration, see https://docs.microsoft.com/en-us/azure/information-protection/verify.

Chapter 3. Tenant key lifecycle management

Operation	Description
Revoke the tenant key	This happens automatically when an organization unsubscribes from Azure RMS. ℹ This may result in loss of access to content protected via Azure RMS and the tenant key.
Refresh the tenant key	Instructions from Generate the tenant key and later sections can be used to create a new tenant key and transfer it securely to Azure RMS. The old tenant key is retained so that earlier content protected by Azure RMS can still be accessed.
Backup and recover the tenant key	The organization is responsible for ensuring that its copy of the tenant key is kept securely and is appropriately backed up. Using the local copy of the key (protected in the organizations Security World) is the only way to retrieve the key. Azure RMS holds a copy of the Tokenized Key Blob that is used for recovery purposes within Azure if necessary (for example, if a node fails.) The version of the key held within Azure RMS cannot be exported.
Export the tenant key	This is not possible from Azure RMS.

Chapter 4. Glossary

Tenant key

A cryptographic key that is unique to an organization and is used as a cryptographic root of trust. The tenant key is used to secure all Rights Management cryptographic functions being undertaken by an organization within a cloud provisioned service. This helps to protect the organizations data from unauthorized third parties, including the cloud service provider.