WINDOWS CERTIFICATE STORE

Integration Guide

Applicable Devices:

Vectera Plus

TABLE OF CONTENTS

[1] DOCUMENT INFORMATION

[1.1] DOCUMENT OVERVIEW

The purpose of this document is to provide information regarding the configuration of the Futurex Vectera Plus HSM with a Windows Certificate Store using the Futurex CNG cryptographic library. For additional questions related to your HSM, see the relevant user guide.

[1.2] ABOUT WINDOWS CERTIFICATE STORES

From Microsoft's documentation website: "On a computer that has the Windows operating system installed, the operating system stores a certificate locally on the computer in a storage location called the certificate store. A certificate store often has numerous certificates, possibly issued from a number of different certification authorities (CAs)."

[1.3] GUARDIAN INTEGRATION

The Guardian Series 3 introduces mission-critical viability to core cryptographic infrastructure, including:

- Centralize device management
- Eliminates points of failure
- Distribute transaction loads
- Group-specific function blocking
- User-defined grouping systems

Please see applicable guide for configuring HSMs with the Guardian Series 3.

[2] PREREQUISITES

Supported Hardware:

- Vectera Plus, 6.7.x.x and above

Supported Operating Systems:

- Windows 2012 R2 (6.3.9600) and above

Other:

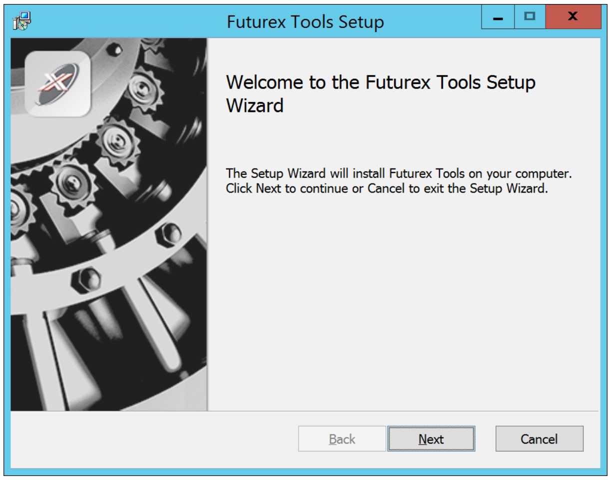
- OpenSSL

[3] INSTALL FUTUREX CNG AND FXCLI USING FXTOOLS

The easiest way to install the Futurex CNG (FXCNG) module and Futurex Command Line Interface (FXCLI) in a Windows environment is with Futurex Tools (FXTools). You can download FXTools from the Futurex Portal. Step-by-step installation instructions are provided below.

Note: Install the FXCNG module on the same computer as the application integrating with the Vectera Plus HSM. Install FXCLI on the workstation you will use to configure the HSM.

Run the Futurex Tools installer as an administrator and follow the prompts to complete the installation.



Welcome to the Futurex Tools Setup Wizard

The Setup Wizard will install Futurex Tools on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

FIGURE: FUTUREX TOOLS SETUP WIZARD

The Setup Wizard installs all tools on the system by default. You can override the defaults and choose not to install certain modules. The installation provides the following services:

- **Futurex Client Tools –** Command Line Interface (CLI) and associated SDK for both Java and C.
- **Futurex CNG Module –** The Microsoft Next Generation Cryptographic Library.
- **Futurex Cryptographic Service Provider (CSP) –** The legacy Microsoft cryptographic library.
- **Futurex EKM Module –** The Microsoft Enterprise Key Management library.
- **Futurex PKCS #11 Module –** The Futurex PKCS #11 library and associated tools.
- **Futurex Secure Access Client –** A client used to connect a Futurex Excrypt Touch to a local laptop through USB, which can then connect to a remote Futurex device.

If the Futurex Secure Access Client was selected, the process will also install the Futurex Excrypt Touch driver, which might start minimized or in the background.

After the installation completes, all services are installed in the C:\Program Files\Futurex\ directory. The CNG Module, CSP Module, EKM Module, and PKCS #11 Module all require configuration files, which are located in their corresponding directory with a .cfg extension. In addition, the installation registers the CNG and CSP Modules in the Windows Registry (HKEY_LOCAL_ MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider), and installs them in the C:\Windows\System32\ directory.

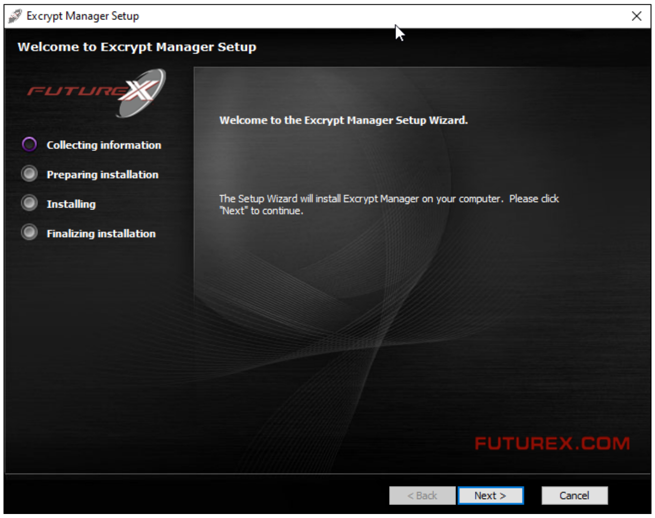
[4] INSTALL EXCRYPT MANAGER

Excrypt Manager is a Windows application that provides a GUI-based method for configuring the HSM in subsequent sections. Installing Excrypt Manager is optional because you can use FXCLI to perform all necessary HSM configurations.

Note: Install Excrypt Manager on the workstation you will use to configure the HSM. If you plan to use a Virtual HSM for the integration, you must perform all configurations using either FXCLI, the Excrypt Touch, or the Guardian Series 3.

Note: The Excrypt Manager version must be from the 4.4.x branch or later to be compatible with the HSM firmware, which must be 6.7.x.x or later.

To install Excrypt Manager, run the Excrypt Manager installer as an administrator and follow the prompts in the setup wizard to complete the installation.



Welcome to Excrypt Manager Setup

Welcome to the Excrypt Manager Setup Wizard.

The Setup Wizard will install Excrypt Manager on your computer. Please click "Next" to continue.

FIGURE: EXCRYPT MANAGER SETUP WIZARD

The installation wizard prompts you to specify where you want to install Excrypt Manager. The default location is C:\Program Files\Futurex\Excrypt Manager\. After choosing a location, select **[Install]**.

[5] CONFIGURE THE FUTUREX HSM

In order to establish a connection between the CNG library and the Futurex HSM, a few configuration items need to first be performed, which are the following:

NOTE: All of the steps in this section can be completed through either Excrypt Manager or FXCLI (if using a physical HSM rather than a virtual HSM). Optionally, steps 4 through 6 can be completed through the Guardian Series 3 (Please refer to the applicable guide for configuring HSMs with the Guardian Series 3).

1. Connect to the HSM via the front USB port (**NOTE**: If you are using a virtual HSM for the integration you will have to connect to it over the network either via FXCLI, the Excrypt Touch, or the Guardian Series 3)
    a. Connecting via Excrypt Manager
    b. Connecting via FXCLI
2. Validate the correct features are enabled on the HSM
3. Setup the network configuration
4. Load the Futurex FTK
5. Configure a Transaction Processing connection and create a new Application Partition
6. Create a new Identity that has access to the Application Partition created in the previous step
7. Configure TLS Authentication. There are two options for this:
    a. Enabling server-side authentication
    b. Creating client certificates for mutual authentication

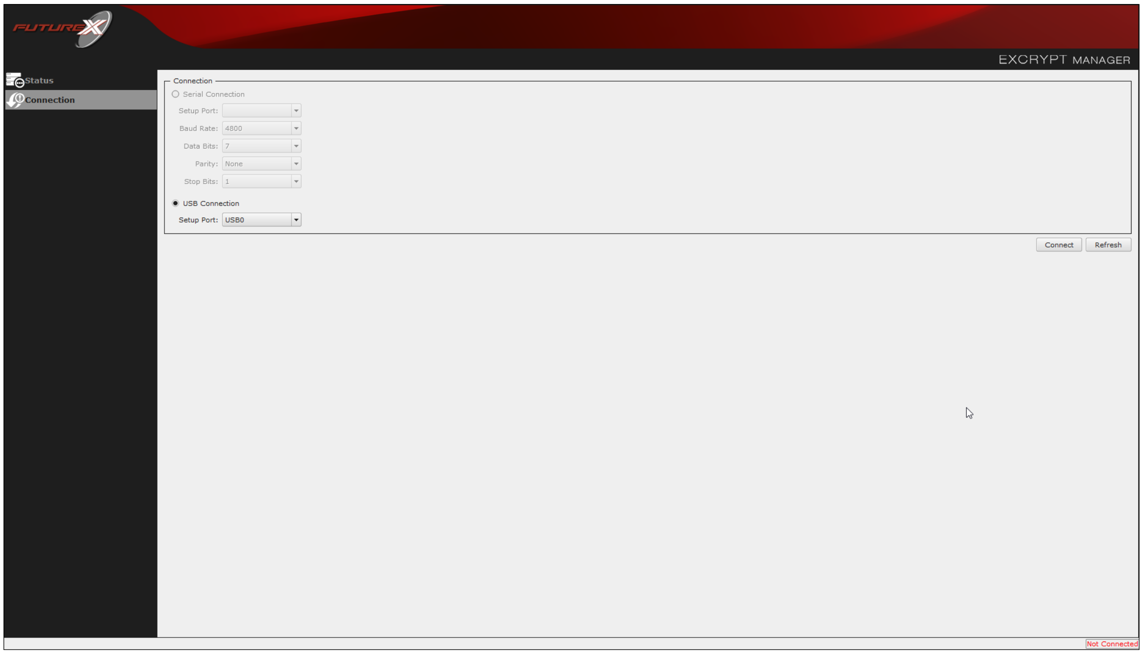
Each of these action items is detailed in the following subsections.

[5.1] CONNECT TO THE HSM VIA THE FRONT USB PORT

For both Excrypt Manager and FXCLI you need to connect your laptop to the front USB port on the HSM.

Connecting via Excrypt Manager

Open Excrypt Manager, click "Refresh" in the lower right-hand side of the Connection menu. Then select "USB Connection" and click "Connect".



○ Serial Connection

Setup Port:

Baud Rate: 4800

Data Bits: 7

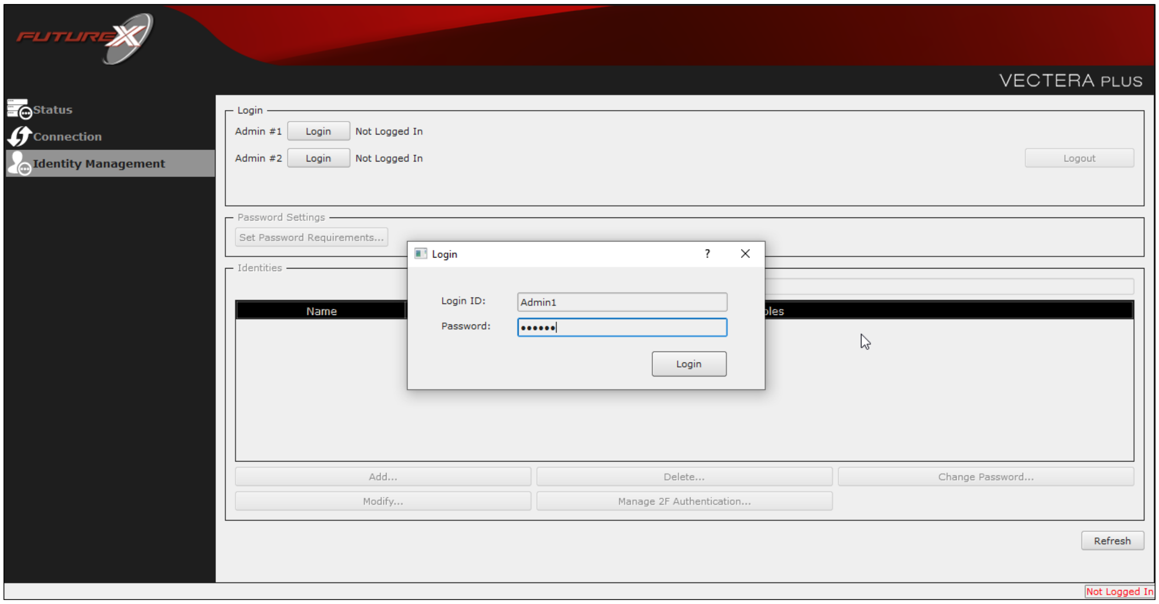
Parity: None

Stop Bits: 1

● USB Connection

Setup Port: USB0

Log in with both default Admin identities.



Admin #1 Login Not Logged In

Admin #2 Login Not Logged In

Login

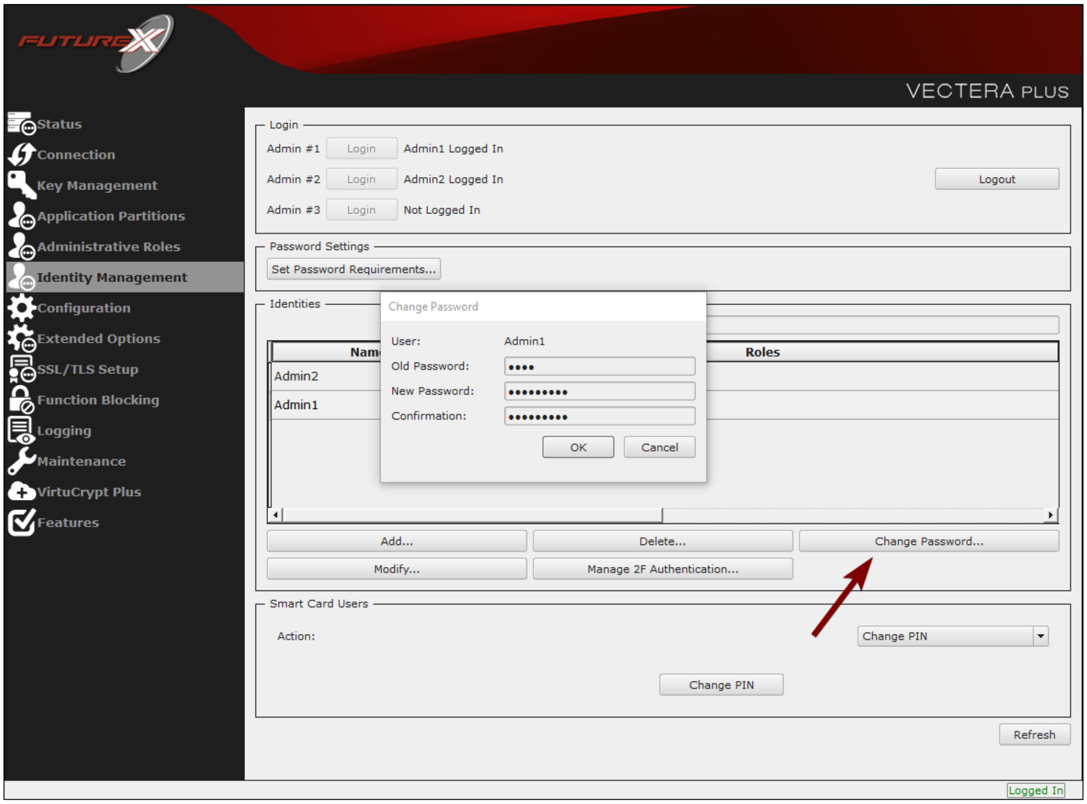
Login ID: Admin1

Password: ••••••

Login

The default Admin passwords (i.e. "safe") must be changed for both of your default Admin Identities (e.g. "Admin1" and "Admin2") in order to load the major keys onto the HSM.

To do so via Excrypt Manager navigate to the Identity Management menu, select the first default Admin identity (e.g. “Admin1”), then click the “Change Password...” button. Enter the old password, then enter the new password twice, and click “OK”. Perform the same steps as above for the second default Admin identity (e.g. “Admin2”).



Admin #1 Login Admin1 Logged In
Admin #2 Login Admin2 Logged In
Admin #3 Login Not Logged In

Admin2
Admin1

User: Admin1
Old Password: ••••
New Password: •••••••••
Confirmation: •••••••••

Connecting via FXCLI

Open the FXCLI application and run the following commands:

```
$ connect usb
$ login user
```

NOTE: The **"login"** command will prompt for the username and password. You will need to run it twice because you must login with both default Admin identities.

The default Admin passwords (i.e. “safe”) must be changed for both of your default Admin Identities (e.g. “Admin1” and “Admin2”) in order to load the major keys onto the HSM.

The following FXCLI commands can be used to change the passwords for each default Admin Identity.

```
$ user change-password -u Admin1
$ user change-password -u Admin2
```

NOTE: The user change-password commands above will prompt you to enter the old and new passwords. It is necessary to run the command twice (as shown above) because the default password must be changed for both default Admin identities.

[5.2] FEATURES REQUIRED IN HSM

In order to establish a connection between the CNG Library and the Futurex HSM, the HSM must be configured with the following features:

- **PKCS #11** -> Enabled
- **Command Primary Mode** -> General Purpose (GP)

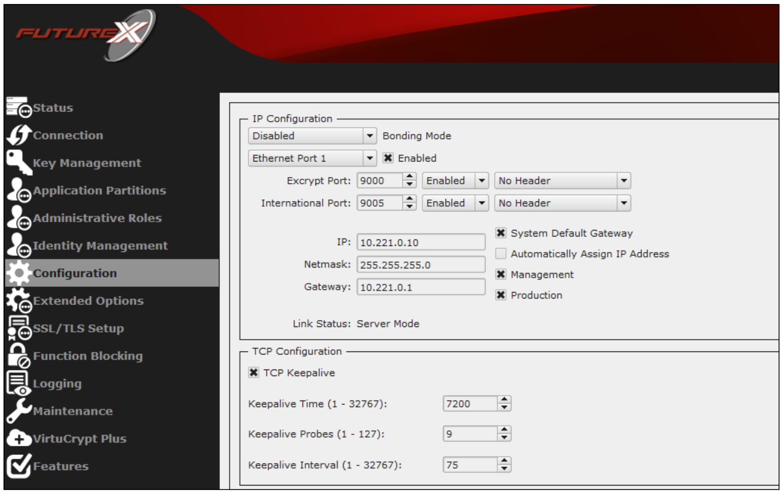
NOTE: For additional information about how to update features on your HSM, please refer to your HSM Administrator's Guide, section **"Download Feature Request File"**.

NOTE: Command Primary Mode = General Purpose, will enable the option to create the FTK major key in the HSM. This key will be required to be able to use the CNG library to communicate with the HSM. For detailed information about how to load major keys in HSMs please refer to your HSM Administrator's Guide.

[5.3] NETWORK CONFIGURATION (HOW TO SET THE IP OF THE HSM)

For this step you will need to be logged in with an identity that has a role with permissions ***Communication:Network Settings****. The default Administrator role and Admin identities can be used.*

Navigate to the *Configuration* page. There you will see the option to modify the IP configuration, as shown below:



IP Configuration

Disabled Bonding Mode

Ethernet Port 1 ☒ Enabled

Excrypt Port: 9000 Enabled No Header

International Port: 9005 Enabled No Header

IP: 10.221.0.10

Netmask: 255.255.255.0

Gateway: 10.221.0.1

☒ System Default Gateway

☐ Automatically Assign IP Address

☒ Management

☒ Production

Link Status: Server Mode

TCP Configuration

☒ TCP Keepalive

Keepalive Time (1 - 32767): 7200

Keepalive Probes (1 - 127): 9

Keepalive Interval (1 - 32767): 75

Alternatively, the following **FXCLI** command can be used to set the IP for the HSM:

```
$ network interface modify --interface Ethernet1 --ip 10.221.0.10 --netmask 255.255.255.0 --gateway
10.221.0.1
```

NOTE: The following should be considered at this point:

- All of the remaining HSM configurations in this section can be completed using the Guardian Series 3 (please refer to Appendix A for instructions on how to do so), with the exception of the final subsection that covers how to create connection certificates for mutual authentication.
- If you are performing the configuration on the HSM directly now, but plan to add the HSM to a Guardian later, it may be necessary to synchronize the HSM after it is added to a Device Group on the Guardian.
- If configuration through a CLI is required for your use-case, then you should manage the HSMs directly.

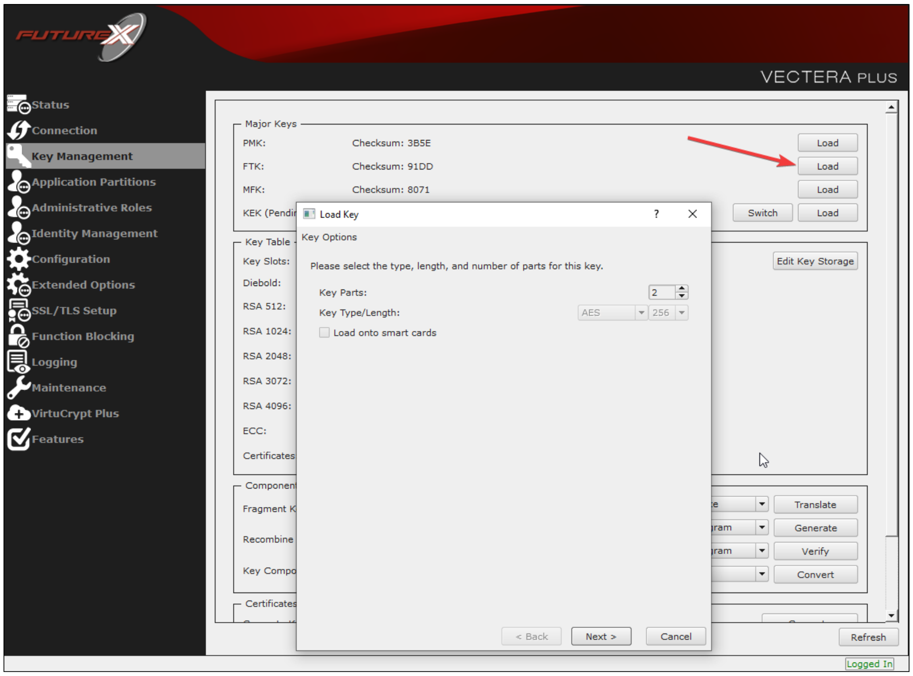
[5.4] LOAD FUTUREX KEY (FTK)

For this step you will need to be logged in with an identity that has a role with permissions ***Major Keys:Load****. The default Administrator role and Admin identities can be used.*

The FTK is used to wrap all keys stored on the HSM used with CNG. If using multiple HSMs in a cluster, the same FTK can be used for syncing HSMs. Before an HSM can be used with CNG, it must have an FTK.

NOTE: This process can also be completed using FXCLI, the Excrypt Touch, or the Guardian Series 3. For more information about how to load the FTK into an HSM using these tools/devices, please see the relevant Administrative Guide.

After logging in, select *Key Management*, then "Load" under FTK. Keys can be loaded as components that are XOR'd together, M-of-N fragments, or generated. If this is the first HSM in a cluster, it is recommended to generate the key and save to smart cards as M-of-N fragments.



Major Keys
PMK: Checksum: 3B5E
FTK: Checksum: 91DD
MFK: Checksum: 8071

Load Key
Key Options
Please select the type, length, and number of parts for this key.
Key Parts: 2
Key Type/Length: AES 256
□ Load onto smart cards

Alternatively, the following **FXCLI** commands can be used to load an FTK onto an HSM.

If this is the first HSM you are setting up you will need to generate a random FTK. Optionally, you can also load it onto smart cards simultaneously with the -m and -n flags.

```
$ majorkey random --ftk -m [number_from_2_to_9] -n [number_from_2_to_9]
```

If it's a second HSM that you're setting up in a cluster then you will load the FTK from smart cards with the following command:

```
$ majorkey recombine --key ftk
```

[5.5] CONFIGURE A TRANSACTION PROCESSING CONNECTION AND CREATE AN APPLICATION PARTITION

*For this step you will need to be logged in with an identity that has a role with permissions **Role:Add, Role:Assign All Permissions, Role:Modify, Keys:All Slots**, and **Command Settings:Excrypt**. The default Administrator role and Admin identities can be used.*

NOTE: For the purposes of this integration guide you can consider the terms "Application Partition" and "Role" to be synonymous. For more information regarding Application Partitions, Roles, and Identities, please refer to the relevant Administrator's guide.

Configure a Transaction Processing Connection

Before an application logs in to the HSM with an authenticated user, it first connects via a “Transaction Processing” connection to the **Transaction Processing** Application Partition. For this reason, it is necessary to take steps to harden this Application Partition. The following three things need to be configured for the Transaction Processing partition:

1. It should not have access to the “All Slots” permissions
2. It should not have access to any key slots
3. Only the CNG communication commands should be enabled

Go to *Application Partitions*, select the Transaction Processing Application Partition, and click Modify.

Under the "Permissions" tab, leave the top-level **Keys** permission checked, but uncheck the **All Slots** sub permission.

Under the "Key Slots" tab you need to ensure that there are no key ranges specified. By default, the Transaction Processing Application Partition has access to the entire range of key slots on the HSM.

Lastly, under the “Commands” tab make sure that only the following **CNG Communication commands** are enabled:

- **ECHO**: Communication Test/Retrieve Version
- **PRMD**: Retrieve HSM restrictions
- **RAND**: Generate random data
- **HASH**: Retrieve device serial
- **GPKM**: Retrieve key table information
- **GPKS**: General purpose key settings get/change
- **GPKR**: General purpose key settings get (read-only)

Alternatively, the following **FXCLI** commands can be used to remove all permissions and key ranges that are currently assigned to the **Transaction Processing** role and enable only the CNG Communication commands:

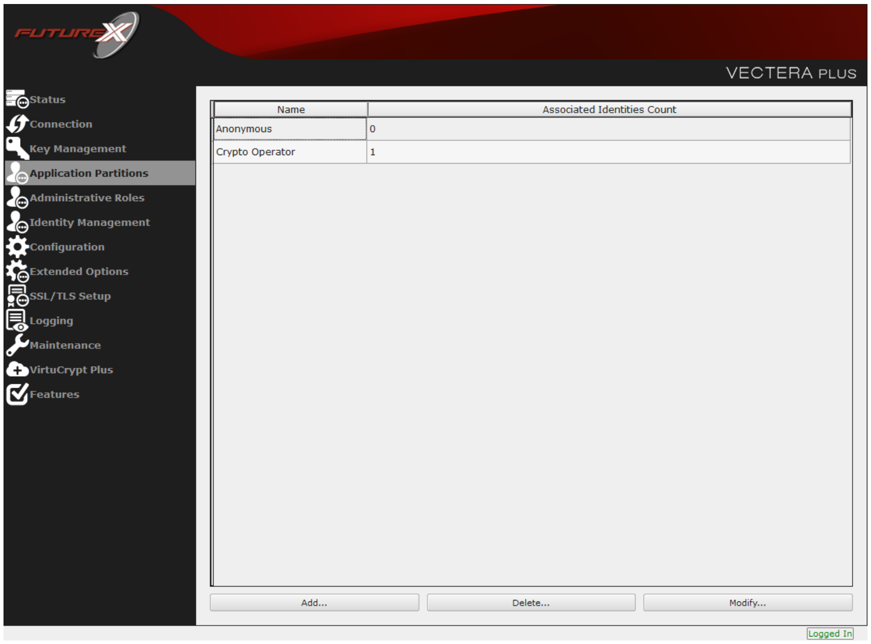
```
$ role modify --name Anonymous --clear-perms --clear-key-ranges
```

```
$ role modify --name Anonymous --add-perm "Keys" --add-perm Excrypt:ECHO --add-perm Excrypt:PRMD --
add-perm Excrypt:RAND --add-perm Excrypt:HASH --add-perm Excrypt:GPKM --add-perm Excrypt:GPKS --
add-perm Excrypt:GPKR
```

Create an Application Partition

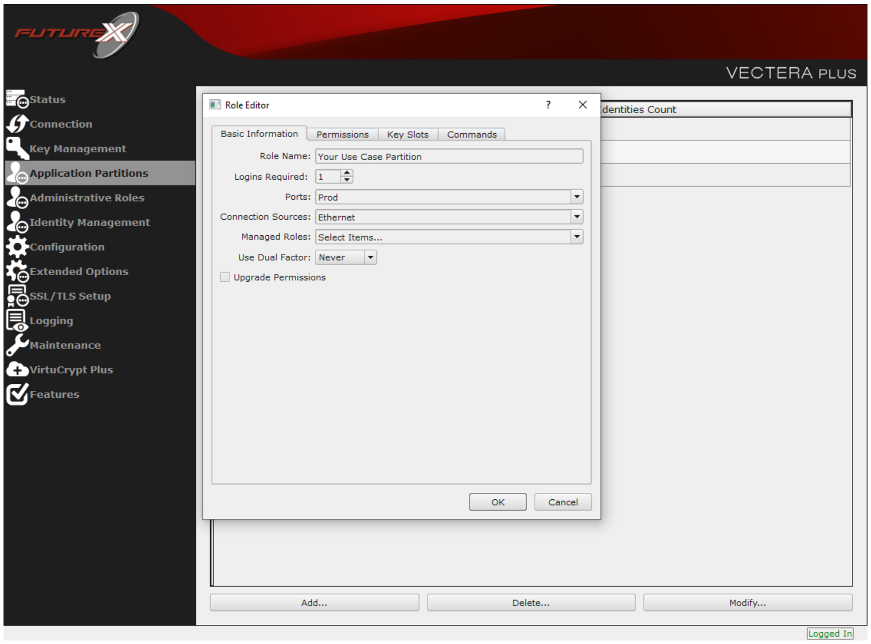
In order for application segregation to occur on the HSM, an Application Partition must be created specifically for your use case. Application partitions are used to segment the permissions and keys on an HSM between applications. The process for configuring a new application partition is outlined in the following steps:

Navigate to the *Application Partitions* page and click the "Add" button at the bottom.



Name	Associated Identities Count
Anonymous	0
Crypto Operator	1

Fill in all of the fields in the *Basic Information* tab exactly how you see below (except for the *Role Name* field). In the *Role Name* field, specify any name that you would like for this new Application Partition. *Logins Required* should be set to "1". *Ports* should be set to "Prod". *Connection Sources* should be configured to "Ethernet". The *Managed Roles* field should be left blank because we'll be specifying the exact Permissions, Key Slots, and Commands that we want this Application Partition/Role to have access to. Lastly, the *Use Dual Factor* field should be set to "Never".



Role Editor

Basic Information | Permissions | Key Slots | Commands

Role Name: Your Use Case Partition

Logins Required: 1

Ports: Prod

Connection Sources: Ethernet

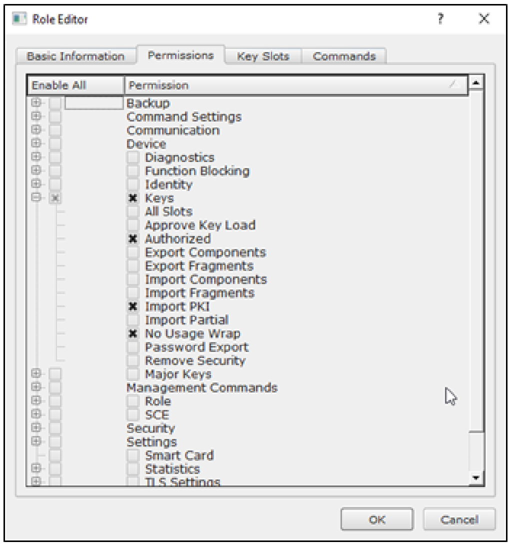
Managed Roles: Select Items...

Use Dual Factor: Never

□ Upgrade Permissions

OK | Cancel

Under the “Permissions” tab, select the key permissions shown in the screenshot below. The **Authorized** permission allows for keys that require login. The **Import PKI** permission allows trusting an external PKI, which is used by some applications to allow for PKI symmetric key wrapping (It is not recommended to enable unless using this use case). The **No Usage Wrap** permission allows for interoperable key wrapping without defining key usage as part of the wrapped key (This is only recommended if exchanging keys with external entities or using the HSM to wrap externally used keys).


Role Editor

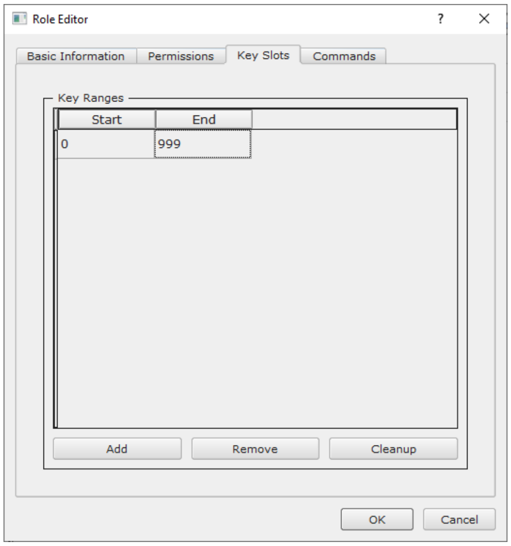
Basic Information | Permissions | Key Slots | Commands

Enable All | Permission

- □ Backup
- □ Command Settings
- □ Communication
- □ Device
  - □ Diagnostics
  - □ Function Blocking
  - □ Identity
  - ☒ Keys
    - □ All Slots
    - □ Approve Key Load
    - ☒ Authorized
    - □ Export Components
    - □ Export Fragments
    - □ Import Components
    - □ Import Fragments
    - ☒ Import PKI
    - □ Import Partial
    - ☒ No Usage Wrap
    - □ Password Export
    - □ Remove Security
  - □ Major Keys
- □ Management Commands
  - □ Role
  - □ SCE
- □ Security
- □ Settings
  - □ Smart Card
  - □ Statistics
  - □ TLS Settings

OK | Cancel

Under key Slots, it is recommended that you create a range of 1000 total keys (here we’ve specified the key range 0-999), which do not overlap with another Application Partition. Within this range, there must be ranges for both symmetric and asymmetric keys. If more keys are required by the application, configure accordingly.



Start	End
0	999

Based on application requirements there are particular functions that need to be enabled on the Application Partition in order to utilize the HSMs functionality. The commands that need to be enabled to for Windows Certificate Store integration are listed below. These can be enabled under the "Commands" tab.

<u>CNG Communication Commands</u>

- **ECHO**: Communication Test/Retrieve Version
- **GPKM**: Retrieve key table information
- **GPKR**: General purpose key settings get (read-only)
- **HASH**: Retrieve device serial

<u>Key Operations Commands</u>

- **APFP**: Generate PKI Public Key from Private Key
- **RPFP**: Get public components from RSA private key

<u>Data Encryption Commands</u>

- **GPSR**: General purpose RSA encrypt/decrypt or sign/verify with recovery

<u>Signing Commands</u>

- **ASYS**: Generate a Signature Using a Private Key
- **RSAS**: Generate a Signature Using a Private Key

Alternatively, the following **FXCLI** commands can be used to create the new Application Partition and enable all of the functions that are needed:

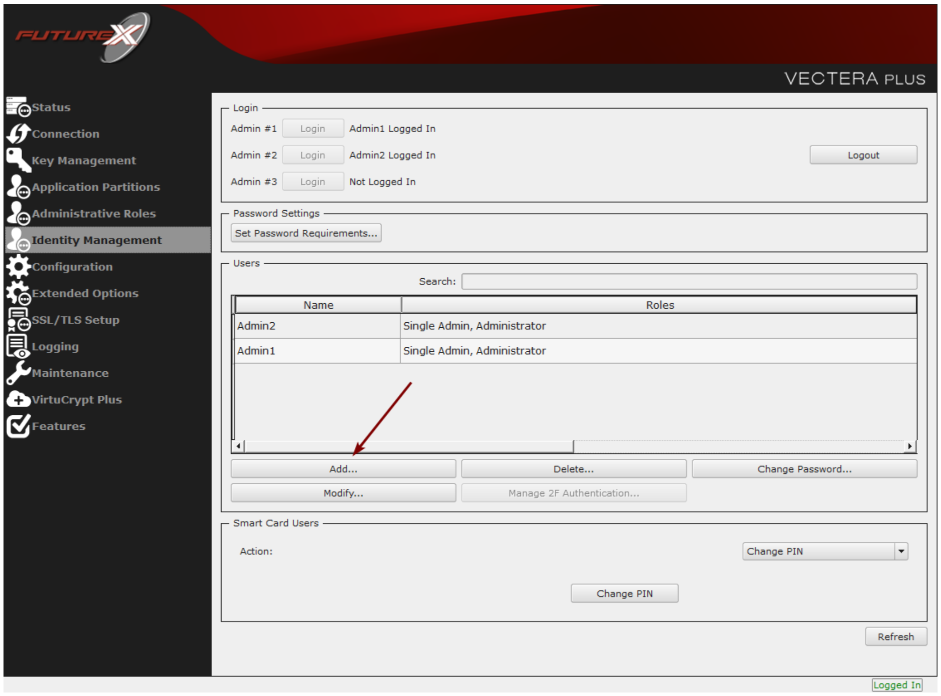
```
$ role add --name Role_Name --application --key-range (0,999) --perm "Keys:Authorized" --perm "Key-
s:Import PKI" --perm "Keys:No Usage Wrap"
```

```
$ role modify --name [role_name] --clear-perms --add-perm Excrypt:ECHO --add-perm Excrypt:GPKM --
add-perm Excrypt:GPKR --add-perm Excrypt:HASH --add-perm Excrypt:APFP --add-perm Excrypt:RPFP --
add-perm Excrypt:GPSR --add-perm Excrypt:ASYS --add-perm Excrypt:RSAS
```

[5.6] CREATE NEW IDENTITY AND ASSOCIATE IT WITH THE NEWLY CREATED APPLICATION PARTITION

*For this step you will need to be logged in with an identity that has a role with permissions **Identity:Add**. The default Administrator role and Admin identities can be used.*

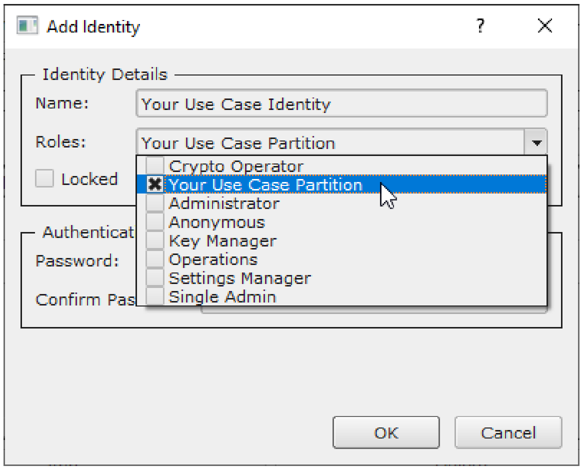
A new identity must be created, which will need to be associated with the Application Partition created in the previous step. To create this new identity, go to *Identity Management*, and click "Add".



Admin #1 Login Admin1 Logged In
Admin #2 Login Admin2 Logged In
Admin #3 Login Not Logged In

Name	Roles
Admin2	Single Admin, Administrator
Admin1	Single Admin, Administrator

Specify a name for the new identity, and in the Roles dropdown select the name of the Application Partition created in the previous step. This will associate the new Identity with the Application Partition that you created.



Name: Your Use Case Identity
Roles: Your Use Case Partition
☐ Locked

Alternatively, the following **FXCLI** command can be used to create a new Identity and associate it with the role that was created:

```
$ identity add --name Identity_Name --role Role_Name --password [password]
```

This new identity must be set in fxcng.cfg file, in the following section:

```
# Identity that is assigned to the created Application Partition
<CRYPTO-OPR>      [insert name of identity that you created]      </CRYPTO-OPR>

# Password of the Identity above
<CRYPTO-OPR-PASS> [password] </CRYPTO-OPR-PASS>

# Production connection
<PROD-ENABLED>    YES         </PROD-ENABLED>
<PROD-PORT>        9100         </PROD-PORT>
```

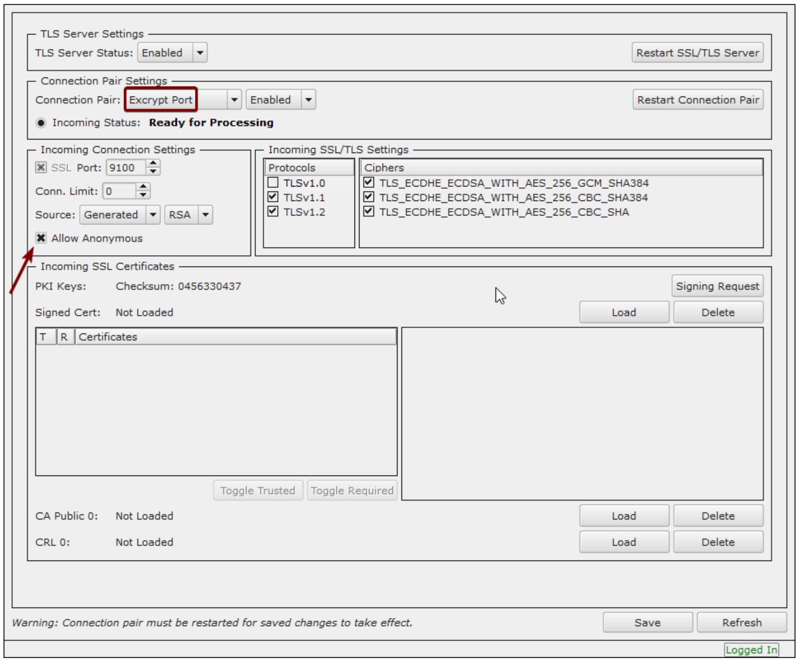
NOTE: Crypto Operator in the fxcng.cfg file must match exactly the name of the identity created in the HSM.

[5.7] CONFIGURE TLS AUTHENTICATION

For this step you will need to be logged in with an identity that has a role with permissions ***Keys:All Slots, Management Commands:Certificates, Management Commands:Keys, Security:TLS Sign****, and* ***TLS Settings:Upload Key****. The default Administrator role and Admin identities can be used.*

Enable Server-Side Authentication (Option 1)

Mutually authenticating to the HSM using client certificates is recommended, but server-side authentication is also supported. To enable server-side authentication go to *SSL/TLS Setup*, then select the Excrypt Port and enable the "Allow Anonymous" setting.



TLS Server Settings

TLS Server Status: Enabled

Connection Pair Settings

Connection Pair: Excrypt Port Enabled

● Incoming Status: **Ready for Processing**

Incoming Connection Settings

☒ SSL Port: 9100

Conn. Limit: 0

Source: Generated RSA

☒ Allow Anonymous

Incoming SSL/TLS Settings

Protocols	Ciphers
☐ TLSv1.0	☑ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
☑ TLSv1.1	☑ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
☑ TLSv1.2	☑ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

Incoming SSL Certificates

PKI Keys: Checksum: 0456330437

Signed Cert: Not Loaded

CA Public 0: Not Loaded

CRL 0: Not Loaded

Warning: Connection pair must be restarted for saved changes to take effect.

Alternatively, the following **FXCLI** command can be used to enable server-side authentication with the "Allow Anonymous" SSL/TLS setting:

```
$ tls-ports set -p "Excrypt Port" --anon
```

Create Connection Certificates for Mutual Authentication (Option 2)

Mutually authenticating to the HSM using client certificates is recommended, and enforced by default. In the example below, FXCLI is utilized to generate a CA that then signs the HSM server certificate and a client certificate. The client keys and CSR are generated in Windows PowerShell with OpenSSL. For other options for managing certificates required for mutual authentication with the HSM, please review the relevant Administrator's guide.

Find the **FXCLI** program that was installed with FXTools, and run it as an administrator.

Things to note:

- For this example, the computer running FXCLI is connected to the front port of the HSM. Remote management is possible however, using the HSMs Web Portal, or the Excrypt Touch.
- For commands that create an output file, if you do not specify a file path (as is the case here) it will save the file to the directory from which the FXCLI program is executed.
- Using user-generated certificates requires a PMK to be loaded on the HSM.
- If you run **help** by itself it will show a full list of available commands. You can see all of the available options for any given command by running the command name followed by **help**.

```
# Connect your laptop to the HSM via the USB port on the front, then run this command.
$ connect usb
```

```
# Log in with both default Admin identities. This command will prompt for the username and pass-
word. You will need to run this command twice.
$ login user
```

```
# Generate TLS CA and store it in an available key slot on the HSM
$ generate --algo RSA --bits 2048 --usage mak --name TlsCaKeyPair --slot next
```

```
# Create root certificate
$ x509 sign \
    --private-slot TlsCaKeyPair \
    --key-usage DigitalSignature --key-usage KeyCertSign \
    --ca true --pathlen 0 \
    --dn 'O=Futurex\CN=Root' \
    --out TlsCa.pem
```

```
# Generate the server keys for the HSM
$ tls-ports request --pair "Excrypt Port" --file production.csr --pki-algo RSA
```

```
# Sign the server CSR with the newly created TLS CA
$ x509 sign \
    --private-slot TlsCaKeyPair \
    --issuer TlsCa.pem \
    --csr production.csr \
    --eku Server --key-usage DigitalSignature --key-usage KeyAgreement \
    --ca false \
    --dn 'O=Futurex\CN=Production' \
    --out TlsProduction.pem
```

```
# Push the signed server PKI to the production port on the HSM
$ tls-ports set --pair "Excrypt Port" \
    --enable \
    --pki-source Generated \
    --clear-pki \
    --ca TlsCa.pem \
    --cert TlsProduction.pem \
    --no-anon
```

***NOTE**: The following OpenSSL commands will need to be run from Windows PowerShell, rather than from the FXCLI program.*

```
# Generate the client keys
$ openssl genrsa -out privatekey.pem 2048
```

```
# Generate client CSR
$ openssl req -new -key privatekey.pem -out ClientPki.csr -days 365
```

Using FXCLI, sign the CSR that was just generated using OpenSSL.

```
# Sign the client CSR under the root certificate that was created
$ x509 sign  \
--private-slot TlsCaKeyPair \
 --issuer TlsCa.pem \
 --csr ClientPki.csr \
 --eku Client --key-usage DigitalSignature --key-usage KeyAgreement \
 --dn 'O=Futurex\CN=Client' \
  --out SignedPki.pem
```

Switch back to Windows PowerShell for the remaining commands.

```
# Use OpenSSL to create a PKCS#12 file that can be used to authenticate, as a client, using our CNG
library
$ openssl pkcs12 -export -inkey privatekey.pem -in SignedPki.pem -certfile TlsCa.pem -out PKI.p12
```

[6] EDIT THE FUTUREX CNG CONFIGURATION FILE

The Futurex CNG configuration file (i.e., `fxcng.cfg`) allows the user to set the Futurex CNG library to connect to the HSM. To edit, run a text editor as an administrator and edit the configuration file accordingly.

NOTE: The Futurex CNG library expects the FXCNG config file to be in a certain location (`C:\Program Files\Futurex\fxcng\fxcng.cfg`), but that location can be overwritten using an environment variable (`FXCNG_CFG`).

[6.1] DEFINING CONNECTION AND AUTHENTICATION DETAILS

Connection and authentication details must be defined in the **<HSM>** section of the FXCNG configuration file.

```
<HSM>
    # Which PKCS11 slot
    <SLOT>                  0                       </SLOT>
    <LABEL>                 Futurex                 </LABEL>

    # HSM crypto operator user name
    <CRYPTO-OPR>            [identity_name]             </CRYPTO-OPR>
    # Automatically login on session open
    <CRYPTO-OPR-PASS>       [identity_password]        </CRYPTO-OPR-PASS>

    # Connection information
    <ADDRESS>               10.0.8.30     </ADDRESS>
    <PROD-PORT>             9100                    </PROD-PORT>
    <PROD-TLS-ENABLED>      YES                     </PROD-TLS-ENABLED>
    <PROD-TLS-ANONYMOUS>    NO                      </PROD-TLS-ANONYMOUS>
#    <PROD-TLS-CA>           /home/user/tls/root.pem        </PROD-TLS-CA>
#    <PROD-TLS-CA>           /home/user/tls/sub1.pem     </PROD-TLS-CA>
#    <PROD-TLS-CA>           /home/user/tls/sub2.pem     </PROD-TLS-CA>
    <PROD-TLS-KEY>          /home/user/tls/PKI.p12       </PROD-TLS-KEY>
    <PROD-TLS-KEY-PASS>     safest                  </PROD-TLS-KEY-PASS>

    # YES = This is communicating through a Guardian
    <FX-LOAD-BALANCE>       NO                      </FX-LOAD-BALANCE>
</HSM>
```

In the **<SLOT>** and **<LABEL>** fields we specify to use PKCS11 slot 0 and assign it the label "Futurex".

In the **<CRYPTO-OPR>** field, the name of the identity that was created for the application partition needs to be specified.

In the **<CRYPTO-OPR-PASS>** field, the password of the identity specified in the **<CRYPTO-OPR>** field needs to be set to log the application into the HSM automatically. CNG does not support logging in through the API, so having the ability to log in using the configuration file allows the application to segment out keys on the HSM by associating the Identity with a specific application partition.

In the **<ADDRESS>** field, the IP of the HSM that the CNG library will connect to is specified.

The **<PROD-PORT>** field declares that the CNG library will connect to Production port 9100.

The **<PROD-TLS-ANONYMOUS>** field defines whether the CNG library will be authenticating to the server or not.

The **<PROD-TLS-KEY>** field defines the location of the client private key. Supported formats for the TLS private key are PKCS #1 clear private keys, PKCS #8 encrypted private keys, or a PKCS #12 file that contains the private key and certificates encrypted under the password specified in the **<PROD-TLS-KEY-PASS>** field.

Because a PKCS #12 file is defined in the **<PROD-TLS-KEY>** field in this example, it is not necessary to define the signed client cert with the **<PROD-TLS-CERT>** tag, or the CA cert/s with one or more instances of the **<PROD-TLS-CA>** tag.

If a Guardian is being used to manage HSMs in a cluster, the **<FX-LOAD-BALANCE>** field must be defined as “YES”. If a Guardian is not being used it should be set to “NO”.

[6.2] SPECIAL DEFINE REQUIRED FOR THIS INTEGRATION

The following define must be added to the **<CONFIG>** section of the FXCNG configuration file:

```
<LOGOUT-ON-SESSION-CLOSE>   NO                 </LOGOUT-ON-SESSION-CLOSE>
```

[7] VERIFY THAT THE FUTUREX CNG MODULE IS PROPERLY CONFIGURED

1. Execute the following command in a command prompt:

```
certutil -csptest -csp "Futurex CNG" RSA
```

2. The module is installed properly if you see the following text:

```
Provider Name: Futurex CNG
      Name:  Provider Module:
      UM(1): fxcng.dll
      0(1): 10001, 1
        0: KEY_STORAGE
...
      Name:  Signature Algorithms:
   RSA
    BCRYPT_ASYMMETRIC_ENCRYPTION_INTERFACE -- 3
    NCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION -- 4
    NCRYPT_SIGNATURE_OPERATION -- 10 (16)

    NCryptCreatePersistedKey(Futurex CNG, RSA)
...
  All Algorithms:
    RSA

CertUtil: -csptest command completed successfully.
```

If you do not see the above text, the module is not installed or configured correctly. Review the logs for additional information. The location of the log file is defined in the FXCNG configuration file in the previous step.

[8] GENERATING A KEY PAIR AND CERTIFICATE ON THE VECTERA PLUS

In this section, Futurex Command Line Interface (FXCLI) will be used to do the following:

1. Create a new key pair on the Vectera Plus
2. Generate a Certificate Signing Request (CSR) from that key pair
3. Sign the CSR using a Certificate Authority (CA) that will also be created on the HSM

[8.1] CONNECT AND LOG IN TO THE HSM VIA FXCLI

1. Run the FXCLI application.
2. Configure TLS certificates for communication between FXCLI and the HSM using the `tls` set of commands.

   NOTE: Run `tls help` to access syntax documentation.
3. Connect to the HSM using the following command:

   ```
   $ connect tcp --connect hsm_ip:9009
   ```
4. Log in to the HSM with the default "Admin1" and "Admin2" identities by running the following command twice (each time it will prompt for username and password):

   ```
   $ login user
   ```

[8.2] CREATE A NEW KEY PAIR ON THE VECTERA PLUS

1. Create a new key pair in the next available key slot on the HSM:

   ```
   $ generate --algo RSA --bits 2048 --name IgDemoKeyPair --usage sign,verify --slot next
   ```

   NOTE: Modify the key usage values to match your specific requirements.
2. Confirm which key slot the private key was added to:

   ```
   $ keytable list
   ```
3. Assign a PKCS11 label to the key (**certutil** needs this external data field to be set so that it can find the key in section 9.2):

   NOTE: The PKCS11 label value should match the name that was set for the key pair in the `generate` command.

   ```
   $ keytable extdata --slot 0 --p11-attr label --p11-value IgDemoKeyPair
   ```

[8.3] GENERATE A CERTIFICATE SIGNING REQUEST (CSR)

1. Generate a Certificate Signing Request (CSR) from the new key pair that was created:

   ```
   $ x509 req --private-slot IgDemoKeyPair --out IgDemo.csr
   ```

[8.4] CREATE A CERTIFICATE AUTHORITY

1. Create a new key pair in the next available key slot on the HSM:

```
$ generate --algo RSA --bits 2048 --usage mak --name CaKeyPair --slot next
```

2. Create a certificate from the new key pair that was created:

```
$ x509 sign --private-slot CaKeyPair --key-usage DigitalSignature --key-usage KeyCertSign --ca
true --pathlen 0 --dn 'O=Futurex\CN=Root' --out Ca.pem
```

   Note that the CA certificate was output to a file called `Ca.pem`.

3. Confirm which key slot the private key was added to:

```
$ keytable list
```

4. Assign a PKCS11 label to the key (**certutil** needs this external data field to be set so that it can find the key in section 9.2):

   NOTE: The PKCS11 label value should match the name that was set for the key pair in the `generate` command.

```
$ keytable extdata --slot 1 --p11-attr label --p11-value CaKeyPair
```

[8.5] SIGN THE CSR USING THE CERTIFICATE AUTHORITY

1. Sign the CSR that was created in section 8.3 using the CA certificate created in section 8.4:

```
$ x509 sign --csr IgDemo.csr --issuer Ca.pem --private-slot CaKeyPair --ca false --key-usage
DigitalSignature --key-usage KeyEncipherment --key-usage DataEncipherment --key-usage KeyAgreement
--eku Client --dn 'O=Futurex\CN=IG-Demo' --out IgDemo.pem
```

   NOTE: Modify the key usage values to match your specific certificate requirements.

   Note that the signed leaf certificate was output to a file called `IgDemo.pem`.

[9] IMPORTING CERTIFICATES INTO WINDOWS CERTIFICATE STORE AND ASSOCIATING THEM WITH THE PRIVATE KEYS STORED ON THE VECTERA PLUS

In the following section, the CA and leaf certificates created on the Vectera Plus in the previous section will be imported into the Windows Certificate Store. Once imported, the **certutil** command line utility will be used to associate the certificates with their corresponding private keys stored on the HSM.

[9.1] IMPORT THE CERTIFICATES USING MICROSOFT MANAGEMENT CONSOLE (MMC) AND THE CERTIFICATES SNAP-IN

1. Open Microsoft Management Console by pressing **Windows+R** to open Run, then type "mmc" in the empty box and click **OK**.
2. At the top of the MMC window, select **File** -> **Add/Remove Snap-in**.
3. In the *Add or Remove Snap-ins* window, select **Certificates** and click **Add**.
4. Select the **Computer account** radio button and click **Next**.
5. Select **Local computer** (selected by default) and click **Finish**.
6. Back in the *Add or Remove Snap-ins* window, click **OK**.
7. In the MMC main console, expand the Certificate snap-in.
8. Navigate to the **Personal** -> **Certificates** pane.
9. Right-click within the Certificates panel and select **All Tasks** -> **Import** to start the Certificate Import Wizard.
10. **Local Machine** should be selected as the Store Location. Click **Next** to continue.
11. Click **Browse**, find and select the leaf certificate file (i.e., `IgDemo.pem`), then click **Next**.
12. Leave the default option selected to place all certificates in the **Personal** certificate store, then click **Next**.
13. Review the summary of the selected options, then click **Finish**.

    NOTE: A notification window should pop up stating that the import was successful.
14. Navigate to the **Trusted Root Certificate Authorities** -> **Certificates** pane.
15. Right-click within the Certificates panel and select **All Tasks** -> **Import** to start the Certificate Import Wizard.
16. **Local Machine** should be selected as the Store Location. Click **Next** to continue.
17. Click **Browse**, find and select the CA certificate file (i.e., `Ca.pem`), then click **Next**.
18. Leave the default option selected to place all certificates in the **Trusted Root Certificate Authorities** certificate store, then click **Next**.
19. Review the summary of the selected options, then click **Finish**.

[9.2] ASSOCIATE THE CERTIFICATES WITH THEIR CORRESPONDING PRIVATE KEYS STORED ON THE HSM USING CERTUTIL

1. The serial numbers of both the CA certificate and the leaf certificate need to be noted down for use in the **certutil** commands that follow. To do so, double-click on each of the certificates, navigate to the **Details** tab, and note down the listed serial number value.
2. Open Windows PowerShell or Command Prompt as an administrator.
3. Run the following command to associate the leaf certificate with its corresponding private key stored on the HSM:

   NOTE: Be sure to substitute "serial_number" with the actual certificate serial number value.

   NOTE: "My" represents the **Personal** certificate store.

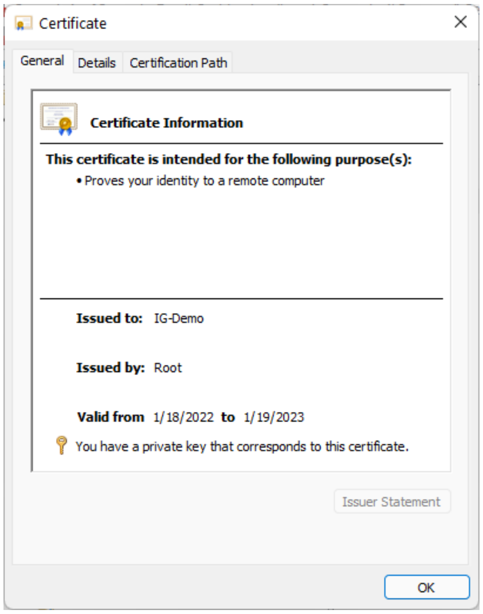
   ```
   $ certutil -repairstore -csp "Futurex CNG" My "serial_number"
   ```

4. Run the following command to associate the CA certificate with its corresponding private key stored on the HSM:

   NOTE: "Root" represents the **Trusted Root Certification Authorities** certificate store.

   ```
   $ certutil -repairstore -csp "Futurex CNG" Root "serial_number"
   ```

5. For further confirmation that both certificates are now associated with their corresponding private keys on the HSM, double-click each of the certificates in the MMC Certificates snap-in and you should now see a message stating that "You have a private key that corresponds to this certificate", as shown below:



Certificate Information

This certificate is intended for the following purpose(s):
- Proves your identity to a remote computer

Issued to: IG-Demo

Issued by: Root

Valid from 1/18/2022 **to** 1/19/2023

You have a private key that corresponds to this certificate.

APPENDIX A: MIGRATING A KEY FROM SOFTWARE STORAGE TO THE VECTERA PLUS

The following appendix will walk through the steps required to migrate a certificate's private key, which is currently stored in software, to a Vectera Plus HSM.

There are two methods that can be used to export a private key from a Windows Certificate Store. 1) Using the MMC Certificates Snap-In, or 2) Using PowerShell commands). Both involve exporting the private key as a PKCS #12 file.

Regardless of which method is used to export the PKCS #12 file from Windows, FXCLI will be the method used to import the private key, contained within the PKCS #12 file, into the Vectera Plus HSM.

NOTE: Before attempting the PKCS #12 export, ensure that the private key of the certificate that is being exported is marked as exportable.

Export the private key From Windows Certificate Store as a PKCS #12 file

Export Method 1: Using the MMC Certificates Snap-In

1. In the MMC Certificates snap-in, right-click the certificate that you wish to export and select **All Tasks** -> **Export** to start the Certificate Export Wizard.
2. In the first dialog, simply click **Next** to continue.
3. Select the **Yes, export the private key** radio button and click **Next**.
4. Select the **Personal Information Exchange - PKCS #12 (.PFX)** radio button (selected by default), and make sure that the **Delete the private key if the export is successful** option is checked. Then, click **Next**.
5. Click the **Password** checkbox, then type-in a password. This will protect the private key in the PKCS #12 file. Click **Next**.
6. Click **Browse**, give the export file a name, select the location where you wish to save it, then click **Next**.

   NOTE: The file extension given to the file must either be .p12 or .pfx.
7. Review the summary of the selected options, then click **Finish**.

   NOTE: A notification window should pop up stating that the export was successful.

Export Method 2: Using PowerShell Commands

1. Open Windows PowerShell as an administrator.
2. Run the following command to determine the **Thumbprint** of the certificate/private key that you want to export:

   ```
   PS C:\>ls Cert:\LocalMachine\My\
   ```

   NOTE: The "My" directory in the file path represents the **Personal** certificate store.

3. Run the the following command to save a password string into the `$mypwd` variable. This will be used as the password for the PKCS #12 file.

   ```
   PS C:\>$mypwd = ConvertTo-SecureString -String "safest" -Force -AsPlainText
   ```

4. Export the PKCS #12 file using the following command:

   ```
   PS C:\>Get-ChildItem -Path Cert:\LocalMachine\My\Thumbprint | Export-PfxCertificate -FilePath
   "C:\Path\To\Desired\Save\Location\file.pfx" -Password $mypwd
   ```

   NOTE: Be sure to substitute "Thumbprint" with the actual thumbprint value of the certificate that you want to export in the `-Path` flag.

5. PowerShell does not provide an option in the `Export-PfxCertificate` command for deleting the private key after successful export of the PKCS #12 file. In order to delete the private key, you must use the `Remove-Item` PowerShell command. This command deletes the certificate as well, though, so the certificate will need to be re-imported afterward.

   First, run the following two commands to export the certificate so that it can later be re-imported:

   ```
   PS C:\>$cert = Get-ChildItem -Path Cert:\LocalMachine\My\Thumbprint

   PS C:\>Export-Certificate -Cert $cert -FilePath "C:\Path\To\Desired\Save\Location\file.cer"
   ```

   Then run the following command to delete the certificate and its private key:

   ```
   PS C:\>Remove-Item -Path Cert:\LocalMachine\My\Thumbprint -DeleteKey
   ```

6. Import the certificate back into the Personal Certificate Store using the following command:

   ```
   PS C:\>Import-Certificate -FilePath "C:\Path\To\Certificate\file.cer" -CertStoreLocation Cer-
   t:\LocalMachine\My
   ```

   NOTE: Be sure to define the actual location of the certificate in the `-FilePath` flag.

Import the PKCS #12 file into the Vectera Plus using FXCLI

1. Run the FXCLI application.

2. Configure TLS certificates for communication between FXCLI and the HSM using the `tls` set of commands.

   NOTE: Run `tls help` to access syntax documentation.

3. Connect to the HSM using the following command:

   ```
   $ connect tcp --connect hsm_ip:9009
   ```

4. Log in to the HSM with the default "Admin1" and "Admin2" identities by running the following command twice (each time it will prompt for username and password):

   ```
   $ login user
   ```

5. Import the PKCS #12 file using the following command:

NOTE: Modify the file path to match the actual location of the PKCS #12 file that you exported from Windows.

```
$ pkcs12 import --file /path/to/pkcs12/file.pfx --slot next --label MigrationDemoKeyPair --
win-system-dacl
```

The command will prompt for the password of the PKCS #12 file. Type the password then press Enter.

NOTE: The above command will import only the private key contained within the PKCS #12 file into the HSM. It will not import the certificate.

6. Confirm which key slot the private key was added to:

```
$ keytable list
```

7. Assign a PKCS11 label to the key (**certutil** needs this external data field to be set so that it can find the key in section 9.2):

   NOTE: The PKCS11 label value should match the value that was set in the `--label` field while importing the PKCS #12 file.

```
$ keytable extdata --slot 3 --p11-attr label --p11-value MigrationDemoKeyPair
```

Re-associate the certificate stored in Windows with the private key stored on the HSM

1. The serial number of the certificate need to be noted down for use in the **certutil** command that follows. To do so, double-click on the certificate in the MMC Certificates snap-in, navigate to the **Details** tab, and note down the listed serial number value.
2. Open Windows PowerShell or Command Prompt as an administrator.
3. Run the following command to associate the certificate with its corresponding private key stored on the HSM:

   NOTE: Be sure to substitute "serial_number" with the actual serial number value of the certificate.

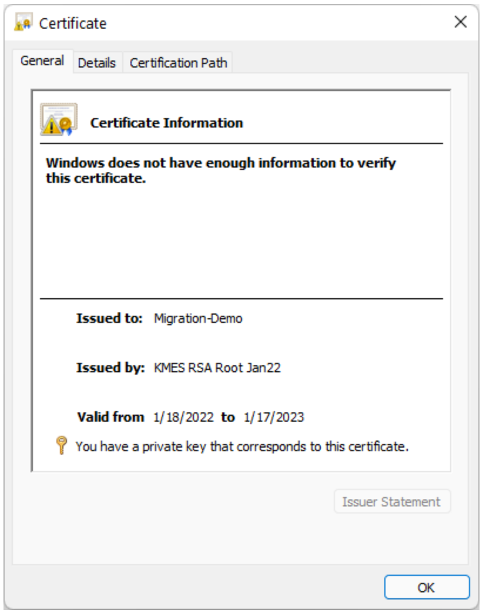
```
$ certutil -repairstore -csp "Futurex CNG" My "serial_number"
```

If the command is successful you will see the following message:

```
CertUtil: -repairstore command completed successfully.
```

4. For further confirmation that the certificate is now associated with its corresponding private key on the HSM, double-click the certificate in the MMC Certificates snap-in and you should now see a message

stating that "You have a private key that corresponds to this certificate", as shown below:



Windows does not have enough information to verify this certificate.

Issued to: Migration-Demo

Issued by: KMES RSA Root Jan22

Valid from 1/18/2022 to 1/17/2023

You have a private key that corresponds to this certificate.

APPENDIX B: XCEPTIONAL SUPPORT

In today’s high-paced environment, we know you are looking for timely and effective resolutions for your mission-critical needs. That is why our Xceptional Support Team does whatever it takes to ensure you have the best experience and support possible. Every time. Guaranteed.

- 24x7x365 mission critical support
- Level 1 to level 3 support
- Extremely knowledgeable subject matter experts

At Futurex, we strive to supply you with the latest data encryption innovations as well as our best-in-class support services. Our Xceptional Support Team goes above and beyond to meet your needs and provide you with exclusive services that you cannot find anywhere else in the industry.

- Technical Services
- Onsite Training
- Virtual Training
- Customized Consulting
- Customized Software Solutions
- Secure Key Generation, Printing, and Mailing
- Remote Key Injection
- Certificate Authority Services

Toll-Free: 1-800-251-5112

E-mail: support@futurex.com

ENGINEERING CAMPUS

864 Old Boerne Road
Bulverde, Texas, USA 78163
Phone: +1 830-980-9782
+1 830-438-8782
E-mail: info@futurex.com

XCEPTIONAL SUPPORT

24x7x365
Toll-Free: 1-800-251-5112
E-mail: support@futurex.com

SOLUTIONS ARCHITECT

E-mail: solutions@futurex.com