YubiHSM 2 for AD#S Guide
Securing Microsoft Active Directory Certificate Services with YubiHSM 2
Copyright
© 2017 Yubico Inc. All rights reserved.

Trademarks
Yubico and YubiKey are registered trademarks of Yubico Inc. All other trademarks are the property of their respective owners.

Disclaimer
The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Yubico shall have no liability for any error or damages of any kind resulting from the use of this document.

The Yubico Software referenced in this document is licensed to you under the terms and conditions accompanying the software or as otherwise agreed between you or the company that you are representing.

Contact Information
Yubico Inc
530 Lytton Ave, Suite 301
Palo Alto, CA 94301
USA
yubi.co/contact

Document Release Date
December 12, 2017
Introduction

This guide is intended to help guide systems administrators successfully deploy YubiHSM 2 with YubiHSM Key Storage Provider. The expected outcome is that the Active Directory Certificate Services CA root key is created securely on the device and that a hardware-based backup copy of key materials has been produced.

This is a guideline for deployment and as such it covers basic topics. Instructions should be modified as required for your specific environment. It is assumed that installation is performed on a single server destined to become a production or lab Certificate Authority root. It is also assumed that you are familiar with the concepts and processes of working with Microsoft Active Directory Certificate Services.

Plan a public key infrastructure (PKI) that is appropriate for your organization. For guidance on setting up a PKI, see Microsoft’s TechNet article on Public Key Infrastructure Design Guidance.

We recommend that you install and test the installation and setup of the YubiHSM 2 in a test or lab environment, before deploying to production.

**Scenario:** In a Windows PKI environment, protect the CA root key in hardware.

**Benefits:** YubiHSM 2 guards the CA root key and protects all signing and verification services using the root key.
Prerequisites and Preparation

The audience of this document is expected to be an experienced system administrator with a good understanding of Windows Server management. In addition, it will be helpful to be familiar with the terminology, software and tools specific to YubiHSM 2. As a primer for these, refer to the Terminology chapter.

In order to follow the steps provided in this guide, be sure you meet the following prerequisites:

- Access to Microsoft Windows Server 2012 R2/2016 with Active Directory in an offline, air-gapped environment, such as a secure computer network that is physically isolated from unsecured networks such as the internet. You must also have elevated system privileges.

- YubiHSM 2 software and tools for Windows have been downloaded from the Yubico Knowledge Base and are available on the system to be used.

- Two (2) factory-reset YubiHSM 2 devices, one for deployment and one for backup in hardware.

- Key custodians have been identified as per local requirements and are available to participate. For more information about the concept of key custodians and the associated ‘m of n’ key shares, see the following section Understanding Key Splitting and Key Custodians.

Installing YubiHSM 2 Tools and Software

You need to download and install the YubiHSM 2 tools and software in order to complete the procedures in this guide.

**TIP:** A generic prompt, $, is used in command line examples in this document. Depending on your command line application, your prompt may be different.

1. Download the package for Windows from YubiHSM Libraries and Tools in the Yubico Knowledge Base. Unzip the archive and move the contents to an appropriate location.

2. On your Windows system, run both installers:

   - yubihsms-cngprovider-windows-amd64.msi (YubiHSM Key Storage Provider)
   - yubihsms-connector-windows-amd64.msi (YubiHSM Connector for Windows)

About the YubiHSM Software

The following YubiHSM pieces of software are used in this guide. They are included as part of the archive file you downloaded from the Yubico Knowledge Base.
## Software Purpose

<table>
<thead>
<tr>
<th>Software</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>YubiHSM Connector</td>
<td>Facilitates communication between the YubiHSM 2 and applications that use it. Must always be running.</td>
</tr>
<tr>
<td>YubiHSM Shell yubihsm-shell.exe</td>
<td>The administrative command line tool used to interact with and configure the YubiHSM 2 device.</td>
</tr>
<tr>
<td>YubiHSM Setup yubihsm-setup.exe</td>
<td>Helps with setting up a device for specific use cases. Currently supports setting up for use with KSP/ADCS.</td>
</tr>
<tr>
<td>YubiHSM Key Storage Provider (KSP)</td>
<td>Acts like a “driver” for the device on Windows and allows it to work with applications that leverage Microsoft’s cryptographic interface, such as Active Directory Certificate Services.</td>
</tr>
</tbody>
</table>

### Default YubiHSM 2 Device Configuration

The YubiHSM 2 device comes with a default, factory-installed authentication key that has the password `password`. As part of the configuration in this guide, the default authentication key will be destroyed. The device can be reset to its default configuration, destroying any objects stored on the device that are not factory-installed. This is done while inserting the device into a USB port -- press the metal rim as you insert the device and continue to press the rim for a minimum of 10 seconds to reset to the default configuration.

#### To verify that the YubiHSM is in default configuration

1. To gain shell access to the YubiHSM 2, launch the YubiHSM Shell program. To do this, launch your command line application, navigate to the directory containing the YubiHSM Shell program, run the command `yubihsm-shell` and press Enter.

2. To connect to the YubiHSM, at the `yubihsm` command line, type `connect` and press Enter. A message verifying that you have a successful connection is displayed.

3. To open a session with the YubiHSM 2, type `session open 1` (where 1 is the id of the default authentication key pre-installed on the device) and press Enter.

4. Type in the default password: `password`
   You will receive a confirmation message that the session has been set up successfully.

5. You now have an administrative connection to the YubiHSM 2 and can list the objects available. To list the objects, type `list objects 0` and press Enter. Your results should be similar to the following:

   ```
   id: 0x0001, type: authkey, sequence: 0
   ```

6. To exit type `quit`. 
Note that you will need two YubiHSM 2 devices to complete all steps of this guide. Specifically, you will deploy a primary device and also create a backup of all key material onto a secondary device.

Understanding Key Splitting and Key Custodians

The preferred method for setting up YubiHSM 2 to secure Microsoft ADCS described in this guide involves key splitting and rejoining, often called setting up an 'm of n' scheme. This involves splitting a wrap key among a predetermined number (n) of wrap key custodians (also known as key shareholders). In order to use the wrap key, a minimum number of shares must be present to rejoin the wrap key (m). If this number is not present, the wrap key cannot be used.

The process ensures no single person can use the wrap key to export key material off the device, and also provides a way to control the import of key material that has been exported under wrap from one device into other devices. The YubiHSM Setup Tool assists in setting up the key splitting and assigning shares to key custodians. If your organization has policies in place that defines how this procedure should be performed, be sure you know these policies before proceeding.

To complete the setup process, you need to know who will have the role of wrap key custodians. You should also have a predetermined practice in place for how the wrap key shares should be recorded (written on paper, photographed, locally printed, or by some other means) and how they are stored between use (for example, offsite archive, safe deposit box, sealed envelope). During setup, all wrap key custodians must be physically present to record, each in turn, their shares. When the wrap key is subsequently used, the shares required to rejoin the wrap key and the key custodians carrying those shares, must also be present at the same time in order to use the wrap key. In other words: 'm' out of 'n' number of shares must be rejoined in order to use the wrap key.

The number, 'n,' should be bigger than 1. However, the number and privacy threshold for key shares ultimately depends on the requirements in your organization.
Configuring the Primary YubiHSM 2 Device

The YubiHSM Setup program, which is part of the YubiHSM 2 toolset, is used to perform the initial configuration of the primary YubiHSM 2 device. This program configures the device with the requirements needed for deploying YubiHSM 2 to safely store the ADCS root CA key. Specifically, during the setup process the YubiHSM is configured so that:

- The necessary key material is generated on the device:
  - One wrap key
  - One application authentication key
  - One audit key
- The wrap key is split among a determined number of key custodians, and each share is recorded by each custodian
- The authentication key and the audit key are exported under wrap to a file, located in the current working directory

To safeguard the integrity of the device, configuration must be performed in an airgap environment.

Configuration Steps

After you have inserted the primary device into the air-gapped system, the configuration steps are as follows. These steps are described in detail in the following procedure.

1. Set up communication between the YubiHSM 2 tools and the device.
2. Start the configuration process.
3. Authenticate to the YubiHSM device.
4. Specify if you need to add RSA decryption (not required if you are using YubiHSM 2 exclusively with ADCS).
5. Enter the domains you need the application authentication key and audit key to be available in.
6. Create the wrap key.
7. Split the wrap key into shares, and record the number of shares required to rejoin the wrap key.
8. Create the application authentication key, which is used to authenticate to the device by the KSP in Windows so the KSP can perform operations in YubiHSM 2.
9. Create the audit key, to access the internal audit log of the device which holds information about the last 62 operations performed and so you can reset the audit log.
To configure your YubiHSM 2 device

1. Enable communication with the YubiHSM 2 device by ensuring that the YubiHSM Connector service (yhconsrv in Windows) is running on the system where the device is inserted. You can validate that the connector is running properly by typing the following URI into your browser:
   http://127.0.0.1:12345/connector/status
   
   Output should be similar to:
   
   status=OK
   serial=* 
   version=1.0.0 
   pid=* 
   address=127.0.0.1 
   port=12345 

2. In your command line application (where $ is the generic prompt used in this document; depending on your command line application, your prompt may be different), run YubiHSM Setup with the argument ksp. To do this, launch your command line application, navigate to the directory containing the YubiHSM Setup program, run the following command, and press Enter.
   $ yubihs-setup ksp
   
   TIP: The setup tool also has a help argument that you can call to learn more about its usage.

3. To start the YubiHSM Setup process, type the default authentication key password: password and press Enter. A confirmation message is displayed that the default authentication key was used and that you successfully have authenticated to the device:
   Using authentication key 0x0001 
   
   Object IDs are displayed in the YubiHSM Setup Tool using hexadecimal numbers, in this case the default authentication key has ID 1, or 0x0001 in hexadecimal.
4. You are prompted if you want to add RSA decryption capabilities. Do one of the following:

- If you plan to use your YubiHSM 2 exclusively with Active Directory Certificate Services (ADCS), you will not need the RSA decryption capabilities, you will only need signing capabilities. Type \( n \).

- If you do need RSA decryption capabilities, if you plan on also using the same YubiHSM 2 device for purposes that do require the capability to decrypt RSA, type \( y \).

**TIP:** If you are unsure what selection to make, select no (\( n \)).

5. The next question to be answered is what domain(s) you need the application authentication key and audit key to be available in (the authentication and audit keys are generated after you create the wrap key). You will only need one domain for the purposes of completing this guide. Do the following:

- Unless you have a requirement to assign more than one domain, type a single number between 1 to 16 and press Enter. In this guide, we assume that domain 1 was entered.

Confirmation will look like the following:

```
got domains [ One ]
```

6. In this step you will create a **wrap key**. The wrap key is very important as it allows you to export and import objects from and to the device. For example, you would export and import objects for backup purposes, as described in the section [Backup Key Material](#). Do one of the following:

- To manually assign a wrap key ID, type the number and press Enter. As object ID ‘1’ is already in use by the default application authentication key, we recommend you assign id ‘2’ to the wrap key. Type 2 and press Enter.

- To allow the system to assign a wrap key ID automatically, type 0 and press Enter.

A confirmation message is displayed:

```
Stored wrap key with ID 0x0002 on the device
```

7. Now you will split the wrap key among a number of key custodians. For this example, we will assume that the wrap key is split into three shares, out of which at least two shares must be present in order to use the key. If there are not two key custodians present, the wrap key cannot be rejoined.

When prompted, do the following:

- Enter the number of shares. In this example, enter 3.

- Enter the privacy threshold. In this example, enter 2.

When defined, the three wrap key custodians will each take their turn in front of the screen to record their respective share. A warning notice appears advising you that the shares are not stored anywhere.
NOTE: Be sure to record the shares and store them safely if you want to re-use the wrap key for this device in the future.

- To start recording the key shares, press Enter.

- The first custodian records his share and leaves the screen. The next one enters and repeats the key share recording for the second share, and so on. Each custodian confirms by pressing y that the share was recorded before handing over to the next. The screen buffer is cleared before each share is presented. Following is an example of a share presented on the screen:

```
2-1-WWmTQj5PHGJQ4H9Y2ouURm8m75QkDOeYzFz0X1VyMpAOeF3YKYZyAVdM0WY4GERclVuAC
```

Have you recorded the key share? (y/n)

It is important to record the whole string presented, including the prefix (in this case) “2-1-” which indicates the number of shares determined to be required to rejoin (or the privacy threshold) and the number of the share itself out of the total number of shares being created.

TIP: For non-production purposes, such as in a lab scenario, it is not necessary to specify that the wrap key should be shared between key custodians but instead just use one solitary key. To do this, when configuring the device using YubiHSM Setup, indicate the number of shares to be 1 and the privacy threshold to also be 1.

8. The setup configuration continues by creating an application authentication key. This key is used to authenticate to the device by the Key Storage Provider (KSP) in Windows, allowing the KSP to perform operations in YubiHSM 2. Since object ID 1 and 2 are already in use by the default authentication key and the wrap key respectively, the example in this guide assumes that the application authentication key to be created next gets ID 3. Do one of the following:

- To manually assign an application authentication key ID, type 3 and press Enter.

- To instead allow the system to assign a wrap key ID automatically, type 0 and press Enter.

You also need to choose a password for the application authentication key. Be sure to store the password of the application authentication key that you will use in a way so that it cannot be compromised. You will need this information later to configure the KSP for use with ADCS. Enter the application authentication key password and press Enter. A confirmation message appears.

```
Stored application authentication key with ID 0x0003 on the device
Saved wrapped application authentication key to {path} 0x0003.yhw
```

The wrapped application authentication key (0x0003.yhw) has been saved to the same path as the location of the YubiHSM Setup program. Although encrypted using the wrap key, we recommend that you do not store keys under wrap on a network-accessible or otherwise potentially compromisable storage media. Leave the file where it was saved for now, as it will be used later to create a backup. You can remove the application authentication key afterwards.

9. The final step of the YubiHSM 2 setup process is to decide whether to create an audit key. The audit key is used to access the internal audit log of the device which holds information about the last 62 operations performed. It is also used to reset the log if needed. Depending on your local requirements, you may not need to create an audit key. If you are unsure of your requirements, we suggest you create an audit key.
When prompted to create an audit key, type `y`. You are then prompted to assign a key ID to the audit key. Be sure to make a note of the ID you enter (for example, key ID 4). You are also prompted to enter the audit key password. Be sure to store this password as well, so that it cannot be compromised. Finally, the audit key will be exported under wrap to the current working directory. Using our example of key ID 4, the file will be named `0x0004.yhw`.

10. The setup tool finishes by letting you know that the default, factory-installed authentication key has been deleted.

```
Previous authentication key 0x0001 deleted
All done
```

Finally, the YubiHSM Setup application exits.

**Verifying the Setup**

You can verify the results of the YubiHSM Setup program by using the YubiHSM Shell program, and logging in using the application authentication key (we used object ID 3 in this guide).

**To verify the YubiHSM Setup**

1. In your command line application (where `$` is the prompt), run YubiHSM Shell program. To do this, if you haven’t already, launch your command line application and navigate to the directory containing the YubiHSM Shell program. Then run the following command and press Enter.

   ```
   $ yubihs-shell
   ```

2. To connect to the YubiHSM, at the `yubihs` prompt, type `connect` and press Enter. A message verifying that you have a successful connection is displayed.

3. To open a session with the YubiHSM 2, type `session open 3` and press Enter.

4. Type in the password for the application authentication key.

   You will receive a confirmation message that the session has been set up successfully.

5. You now have an administrative connection to the YubiHSM 2 and can list the objects available. To list the objects, type `list objects 0` and press Enter. Your results should be similar to the following:

   ```
   Found 3 object(s)
   id: 0x0002, type: wrapkey, sequence: 0
   id: 0x0003, type: authkey, sequence: 0
   id: 0x0004, type: authkey, sequence: 0
   ```

   As you can see by looking at their IDs, these objects correspond to the wrap key, the application authentication key and the audit key that were just created.

6. To obtain more information about any one of the objects, for example, the application authentication key (object ID 3), including its `capabilities`, type the following command and press Enter:

   ```
   yubihs> get objectinfo 0 3 authkey
   ```
The response you receive should look similar to the following:

id: 0x0003, type: authkey, algorithm: yubico-aes-auth, label: "Application auth key", length: 40, domains: 1, sequence: 0, origin: imported, capabilities:
  asymmetric_gen:asymmetric_sign_pkcs:asymmetric_sign_pss:export_wrapped:import_wrapped:export_under_wrap, delegated_capabilities:
  asymmetric_gen:asymmetric_sign_pkcs:asymmetric_sign_pss:export_under_wrap,

This indicates that YubiHSM 2 as it has now been configured will later on allow the KSP to leverage the device to:

- Generate asymmetric objects
- Compute signatures using RSA-PKCS1v1.5
- Compute signatures using RSA-PSS
- Export other objects under wrap
- Import wrapped objects
- Mark an object as exportable under wrap

In addition, this object (the application authentication key, object ID 3) also has so-called delegated capabilities. Delegated capabilities define the set of capabilities that can be set or "bestowed" onto other objects that are created by it.

7. To exit, type **quit**.
Deploying YubiHSM 2 with Active Directory Certificate Services

With a YubiHSM 2 device now configured for use with YubiHSM Key Storage Provider and Microsoft Active Directory Certificate Services, the next set of steps covers the deployment in the ADCS environment. Note that YubiHSM Key Storage Provider software must be installed on the system before proceeding.

Deploying YubiHSM consists of three steps as follows. These steps are described in detail in the following procedure.

1. Configuring the Windows Registry for the YubiHSM Key Storage Provider for the primary YubiHSM 2 device that was configured earlier

2. Configuring ADCS (if not already present)

3. Configuring a new ADCS CA with a root CA key being generated on the device

**Preconditions:**
- Pre-configured primary device
- YubiHSM 2 software installed on air-gap computer
- Windows Server with Active Directory, elevated permissions user

The host that these steps are performed on is assumed to be a member server in the Active Directory domain (domain-joined, not a Domain Controller).

These instructions include steps for a basic configuration and should be performed by an experienced system administrator.

**Configuring the Windows Registry**

For ADCS to use the YubiHSM 2, the following registry entries need to be changed from their default values. The `HKEY_LOCAL_MACHINE\SOFTWARE\Yubico\YubiHSM subkey` was created during installation. Be sure to make a backup of your Registry before you make any changes.

**To configure the Windows Registry**

1. Click **Start > Run**, type `regedit` in the Run dialog box, and click **OK**.
2. Locate and then click the registry subkey for YubiHSM
   (HKEY_LOCAL_MACHINE\SOFTWARE\Yubico\YubiHSM).

3. To change the URI where the connector is listening, change the following entry:
   
   "ConnectorURL"=http://127.0.0.1:12345

4. To change the ID of the application authentication key (object ID 3 was used as an example in this guide; if you used another object ID be sure to enter the correct information). For our example, because the hexadecimal value of 0x00000003 resolves to "3" in the Windows Registry, change the entry as follows:
   
   "AuthKeysetId"=3

5. To change the password for the application authentication key that is stored in the registry change the entry for:
   
   "AuthKeysetPassword"={password}

   The password is stored here for the Key Storage Provider to use when authenticating to the device.

6. To save your changes, exit the Windows Registry.

The YubiHSM Connector service reads the configuration file, yubiHsm-connector-config.yaml. Depending on your local setup, for instance if you are running multiple instances of the software on the same host, you may need to edit this configuration file to make sure that parameters are consistent between the configuration file and the Windows Registry. On Windows, the yubiHsm-connector.config.yaml file is available at C:\programdata\yubiHSM\yubiHsm-connector.yaml - you will need administrator rights to modify the file.

**Setting Up Your Enterprise Certificate Authority**

**To configure ADCS**

If you already have Certification Services installed, you can skip these steps.

1. On a Windows Server host, joined to an existing Active Directory domain, log on into the server as a domain administrator.

2. Click **Start > Administrative Tools**, then click **Server Manager**.

3. Under Roles Summary, click **Add roles and features**.

4. Use the Add Roles and Features Wizard to add the Active Directory Certificate Services role, and click **Next**.

5. In the Select role services wizard page, select the option for **Certification Authority**, then click **Next**.

6. Complete the wizard and reboot the host if prompted.
To configure the ADCS CA and create the Root Key

After you have completed the feature installation, you need to create the Enterprise CA instance.

1. If you haven’t already, do the following:
   - On a Windows Server host, joined to an existing Active Directory domain, log into the server as a domain administrator.
   - Click Start > Administrative Tools, then click Server Manager.
2. In Server Manager, start the Add Roles and Features Wizard and select Role-based or feature-based installation. Click Next.
3. In the Credentials page, confirm that you are logged in as a domain administrator. If you are not, you will not be able to create an Enterprise CA in the subsequent steps. Click Next.
4. In the Role Services page, select the option for Certification Authority, and then click Next.
5. In the Setup Type page, select the option for Enterprise CA, and then click Next.
6. In the CA Type page, select the option for Root CA, and then click Next.
7. In the Private Key page, select the option for Create a new private key, and then click Next.
8. In the Cryptography for CA page, do the following:
   - Click Select a cryptographic provider and select RSA#YubiHSM Key Storage Provider from the list displayed. This indicates that the root key should be generated on the device.
   - Click Key Length and select the key size you want from the list displayed. Options for key size 2048-bit or 4096-bit. The default setting is 2048.
   - For Select the hash algorithm for signing certificates issued by this CA, select a desired hash algorithm, such as SHA256. The default setting is SHA256.
   - Select the option to Allow administrator interaction when the private key is accessed by the CA. This allows the private key to be exported for backup purposes (so it can be restored to another server).
   - Click Next.
9. In the CA Name page, accept the defaults. Click Next.
10. In the Validity Period page, accept the default or set another validity period appropriate for your purposes. Click Next.
11. In the CA Database page, accept the default location for logs. Click Next.
12. In the Confirmation page, the important detail is that the YubiHSM Key Storage Provider is being used to store the CA private key. Click Configure.
13. The Progress page appears, briefly, as the local CA database is created, and changes are written to Active Directory.

14. Finally, confirm the presence of the Configuration succeeded message in the Results page. Click Close.
Back Up Key Material

We strongly recommend you make a backup copy of all production objects residing on your primary device, particularly once the CA root key has been generated on the device. If there is an unforeseen hardware failure of the primary device, having a backup ensures that you can resume operations quickly. In addition, this provides a means to backup all objects contained on a device to reside in secure hardware offline.

Specific recommendations for governance of your critical key material is out of scope for this guide. Make sure to design and document these security procedures to fit the requirements of your organization.

Back Up the YubiHSM 2

The backup of the primary YubiHSM 2 is a duplicate of all of the objects stored on the primary device, to be exported under wrap and that are available using the application authentication key used. For instance, when following this guide, the wrap key (created with ID 2 previously), the application authentication key (ID 3), the audit key (ID 4) (if created previously), and the CA root key will be duplicated onto the secondary device. The factory-installed authentication key (ID 1) on the secondary device will be destroyed. You will need assistance from the wrap key custodians to provide their respective wrap key shares, if applicable. In the example we used in this guide, 2 out of the 3 shares must be available.

When you create a backup, you create a duplicate of the objects on your primary YubiHSM 2 onto a secondary device. The actual backup procedure consists of steps as follows. These steps are described in detail in the following procedure.

1. Set up communication between the YubiHSM 2 tools and the device.
2. Start the configuration process, and authenticate to the YubiHSM 2 device.
3. Identify the CA root key ID.
4. Export the CA root key.
5. Verify the key material under wrap.
6. Restore the key material onto a secondary (backup) device.
7. Verify the objects on the secondary device are correct.
Since the CA root key was created on the device when setting up the CA, it currently only exists on the device. In order to back it up using the YubiHSM Setup program, it must first be exported from the device using the wrap key that also sits on the device alongside the application authentication key and the audit key.

**To export the CA root key under wrap using the wrap key on the device**

1. In your command line application, run YubiHSM Shell program. To do this, if you haven’t already, launch your command line application and navigate to the directory containing the YubiHSM Shell program. Then run the following command and press Enter.

   ```bash
   $ yubihs-m-shell
   ```

2. To connect to the YubiHSM, at the `yubihs` prompt, type `connect` and press Enter. A message verifying that you have a successful connection is displayed.

3. To open a session with the YubiHSM 2, type `session open 3` and press Enter.

4. Type in the password for the application authentication key.

5. You will receive a confirmation message that the session has been set up successfully.

6. If you already know the object ID of the root CA, you can skip this step. If you need to identify the root CA, you can list the objects available. To list the objects, type `list objects 0` (where 0 is the session number) and press Enter.

   You will receive a list of the objects on the device that application authentication key with ID 3 has access to, which will include the CA root key. Identify its ID.

7. To export the CA root key under wrap from the primary device to the local file system, in the YubiHSM Shell program, run

   ```bash
   yubihs> get wrapped 0 2 asymmetric {rootkeyID} rootkey.yhw
   ```

   (Where 0 is the session number, 2 is the wrap key ID, asymmetric is the key type, `{rootkeyID}` is the CA root key with an ID assigned to it by the YubiHSM Key Storage Provider and rootkey.yhw is the name of the file output.) The file rootkey.yhw will be saved to the current working directory.

8. Verify that all the keys that were exported under wrap to file reside in the same directory as the YubiHSM Setup program. The tool looks for files with the `.yhw` file extension in the current working directory and attempts to read and import them into the device. The wrap key will be
imported as a result of providing the wrap key shares to the tool. Given the example object IDs in this guide, the following files should be present:

0x0003.yhw  (Application authentication key under wrap)
0x0004.yhw  (Audit key under wrap)
rootkey.yhw  (CA root key under wrap)

9. To begin the process of restoring the data onto the secondary YubiHSM 2, if the primary YubiHSM 2 device is inserted into your computer, remove it and insert the secondary device. Restoring a device must be performed in an air-gapped environment in order to guarantee integrity.

10. In your command line application (where $ is the prompt), run YubiHSM Setup with the argument restore. To do this, launch your command line application, navigate to the directory containing the YubiHSM Setup program, run the following command, and press Enter.

   $ yubihsm-setup restore

11. To start the YubiHSM Setup process, type the default authentication key password: password and press Enter. A confirmation message is displayed that the default authentication key was used and that you successfully have authenticated to the device:

   Using authentication key 0x0001

   You will now start the restore procedure, which involves providing the number of wrap key shares required by the privacy threshold defined when setting up the primary device.

12. When prompted, type the number of shares required by the privacy threshold and press enter. In this guide, we have specified that 2 shares are required to be rejoined. These must be present in order to proceed.

13. When prompted, for share number 1, the wrap key custodian holding the first share inputs this information and presses Enter. A message is displayed that the share is received:

   Received share 2-1-
   WWmTQj5PHGJQ4H9Y2ouURm8m75QkD0eYzFzOx1VyMpaOeF3YKY2yAVdM0WY4GercLVuAC

14. Continue to have each wrap key custodian enter the share information for each of the wrap key shares required to rejoin the key share. Once the sufficient number of wrap key shares have been inserted by the wrap key custodians, a final message is displayed:

   Stored wrap key with ID 0x0002 on the device

15. Note that the ID of the wrap key on the secondary device is the same as that for the primary device. Once the wrap key has been stored on the secondary device, the YubiHSM Setup program reads the files containing the application authentication key, the CA root key, and, if applicable, the audit key that were saved to file under wrap during the configuration of the primary device.

   reading ./0x0004.yhw
   Successfully imported object Authkey, with ID 0x0004
   reading ./0x0003.yhw
   Successfully imported object Authkey, with ID 0x0003
   reading ./rootkey.yhw
   Successfully imported object Asymmetric, with ID {rootkeyID}
16. If there are files containing wrapped objects with the .yhw file extension in this directory that were exported with a different wrap key than the one reconstituted by the shares here, the setup tool attempts to also read those but will fail gracefully and only restores the files it can decrypt.

17. The restore process finishes and the setup tool lets you know that the default, factory-installed authentication key has been deleted.

   Previous authentication key 0x0001 deleted
   All done

   Finally, the YubiHSM Setup application exits.

**Confirming the Duplicated YubiHSM 2**

You now have a duplicate of the device configured with the three key objects you created on the primary device earlier. These are identical to the primary device that was configured earlier.

**To confirm the duplicated YubiHSM 2**

1. In your command line application, run YubiHSM Shell program. To do this, if you haven’t already, launch your command line application and navigate to the directory containing the YubiHSM Shell program. Then run the following command and press Enter.

   $ yubihsm-shell

2. To connect to the YubiHSM, at the yubihsm prompt, type connect and press Enter. A message verifying that you have a successful connection is displayed.

3. To open a session with the YubiHSM 2, type session open 3 (where 3 is the ID for your application authentication key) and press Enter.

4. Type in the password for the application authentication key. You will receive a confirmation message that the session has been set up successfully.

5. To list the objects, type list objects 0 (or instead of 0 some other session number that was given to you in step 4) and press Enter. Verify that the secondary device now contains all of the key material that you intended to restore.

6. Depending on the order in which the keys under wrap were imported, the order of the enumerated keys on the secondary device may be different than on the primary device when using the list command. This has no practical implication and the object IDs are identical between the devices.

   If you have verified that the secondary device now contains all of the key material that you intended to restore, you should now remove the keys under wrap currently on file in the current working directory for the YubiHSM Setup program.
Alternative Scenarios

This guide only covers basic setup and use of the YubiHSM 2 with ADCS. Some alternative scenarios include migrating an existing CA root key to YubiHSM 2, or leveraging the YubiHSM 2 and YubiHSM Key Storage Provider in larger PKI installations using multiple hosts to serve the CA including Subordinate CAs. Since conditions can vary a great deal between organizations on these topics, the following contains some references that might be useful when deploying YubiHSM 2 under such circumstances.

Migrating an Existing CA Root Key to YubiHSM 2

One potential circumstance when deploying YubiHSM 2 to secure ADCS is the fact that a CA root key already exists, either in software or secured by hardware such as another Hardware Security Module. It is normally possible to migrate the CA root key over to the YubiHSM 2, however depending on the pre-existing setup, the steps to take may vary. For more information, see the information on the Yubico developers’ website at [Move Software Keys to Key Storage Provider](https://developers.yubico.com/yubikey/).

Subordinate CAs

In order to improve security and scalability of your Certification Authority, consider installing the Root CA on a standalone (offline) server, and use a Subordinate CA for all certificate signing.

For additional information about implementing advanced configurations, see the relevant Microsoft documentation on the subject (such as [AD CS Step by Step Guide: Two Tier PKI Hierarchy Deployment](https://docs.microsoft.com/en-us/azure/active-directory/active-directory-pki-adcs-step-bystep-guide-two-tier-pki-hierarchy-deployment)).

Alternative Backup and Restore Procedures

In more advanced installations, such as when the YubiHSM Setup program was not used to set up YubiHSM 2 for ADCS, or when moving the YubiHSM 2 device containing the root CA key from one instance of ADCS to another, see the information on the Yubico developers’ website at [Backup and Restore](https://developers.yubico.com/yubikey/).
Getting Help

Should you require assistance when deploying YubiHSM 2 with ADCS by using this guide, start by referencing the product documentation and currently known issues. If you need additional help, contact Yubico:

- Yubico Knowledge Base
- Product documentation
- Known issues and limitations
- Yubico Forums
- Submit a support request to report an issue
Terminology Used

The following terminology as it relates to YubiHSM 2 is used throughout this guide.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default authentication key</td>
<td>Factory-installed AES key used when initializing the device. Possesses all capabilities.</td>
</tr>
<tr>
<td>Application authentication key</td>
<td>AES key used to authenticate to device. Performs operations according to its defined capabilities.</td>
</tr>
<tr>
<td>Audit key</td>
<td>AES authentication key with rights to access audit log.</td>
</tr>
<tr>
<td>Wrap key</td>
<td>AES key used to protect key material when exporting to file from device and when importing from file to device. Key material exported under wrap will be encrypted and can only be decrypted using the wrap key.</td>
</tr>
<tr>
<td>Capability</td>
<td>A description of what operations are allowed on or with an object such as a key.</td>
</tr>
<tr>
<td>Delegated capability</td>
<td>A description of what operations are allowed on or with an object delegated by the authentication key or wrap key that was used to create it.</td>
</tr>
<tr>
<td>Domain</td>
<td>A logical “container” for objects that can be used to control access to objects on the device.</td>
</tr>
<tr>
<td>Object ID</td>
<td>Object IDs are unique identifiers for any kind of object stored on YubiHSM 2. Their ID can range between 1 and 65535 however the device can hold a maximum of 256 unique objects.</td>
</tr>
<tr>
<td>M of n</td>
<td>Scheme where Wrap key is split into n shares held by key custodians, where at least m shares are needed to use the key (sometimes this is also called 'quorum').</td>
</tr>
<tr>
<td>Key custodian</td>
<td>Holder of a wrap key share.</td>
</tr>
</tbody>
</table>