Integration Guide

PrimeKey EJBCA Enterprise and Luna SA HSM for Government

007-500149-001

Revision B

24-Jun-2019

Please refer to: www.safenetat.com/document-disclaimer for information regarding use and limitations associated with this document.

Table of Contents

Preface **4**

Scope **4**

Technical Support Information **4**

1 Introduction **5**

1.1 Overview 5

1.2 Integration Matrix 5

1.3 Prerequisites 5

1.4 Setup Synopsis 6

2 Verify the HSM Configuration **6**

3 Create the Crypto Tokens and Key Pairs **7**

3.1 Create the Root CA Crypto Token 7

3.2 Create the Root CA Key Pairs 10

3.3 Create the Subordinate CA Crypto Token 11

3.4 Create the Subordinate CA Key Pairs 12

3.5 View the Crypto Tokens and Keys 12

4 Create the Root and Subordinate Certification Authority **13**

4.1 Create the Root Certificate Profile 13

4.2 Create a Subordinate Certificate Profile 16

4.3 Create the Root Certification Authority 19

4.4 Create the Subordinate Certification Authority 24

5 Verify the Keys on the HSM **28**

Preface

Scope

This configuration guide provides instruction for integrating PrimeKey EJBCA Enterprise and the Luna SA for Government, a dedicated Hardware Security Module (HSM) that provides secure generation and storage of private keys.

Technical Support Information

If a problem occurs during installing, registering, or operating this product, please review the documentation. For assistance in resolving the issue, contact the supplier or SafeNet Assured Technologies (SafeNet AT) Customer Support. SafeNet AT Customer Support operates 24 hours a day, 7 days a week. The level of access for this service is governed by the support plan arrangements made between SafeNet AT and the organization. Please consult the support plan for further information about entitlements, including the hours when telephone support is available.

Contact method	Contact information	
Address	SafeNet Assured Technologies, LLC. 3465 Box Hill Corporate Center Drive Suite D Abingdon, MD 21009 USA	
Phone	United States	(866) 307-7233
Web	http://www.safenetat.com/support/	
Support and Downloads	http://www.safenetat.com/support/ Provides access to the SafeNet Assured Technologies Knowledge Base and quick downloads for various products.	

1 Introduction

1.1 Overview

This Integration Guide provides the necessary steps for configuring PrimeKey EJBCA Enterprise to use the SafeNet Luna SA for Government HSM to secure the private keys for its Certification Authorities (CAs) and other components. The guide assumes the Luna client has been on the machine hosting EJBCA Enterprise and that it's configured as detailed in the Luna Client Installation Guide. The Luna client acts as a PKCS #11 provider to EJBCA, and when installed in the default directory, is automatically detected by EJBCA and becomes a selectable PKCS #11 Library option via the GUI.

Instruction is provided for configuring elements of EJBCA with selected security options based on a test environment. Adjust options based on security policy and consult the comprehensive EJBCA Enterprise documentation for more detailed information. Additional PrimeKey documentation on the HSM integration is available here.

The sample integration in this guide includes the creation of two CAs: a Root CA and a Subordinate CA that is acting as an Issuing CA. Each CA requires its own partition on the HSM, and it's assumed during HSM installation that two partitions were created.

1.2 Integration Matrix

This table enumerates all the versions of products tested in this integration.

Platforms Tested	PrimeKey EJBCA	Luna SA
CentOS 7.6 (evaluation virtual machine image provided by PrimeKey)	EJBCA 7.0.1.1 Enterprise (r31723)	Appliance 5.4.7-3 Firmware 6.10.7 Client 5.4.9

1.3 Prerequisites

In order to configure EJBCA to use the Luna SA HSM the following perquisites must be met:

- PrimeKey EJBCA Enterprise has been installed on a server.
- The SafeNet AT Luna HSM is installed and operational with **two** partitions created for EJBCA, one each for the Root and Subordinate CAs.
- The SafeNet AT Luna Client is installed on the server running EJBCA and is installed in the default directory offered during installation.
- The Network Trust Link (NTL) is established between the Luna Client and the Luna HSM. If this has not been done, consult the Luna SA product documentation or the following document:

007-500113-001 - Configuring a Network Trust Link between a Luna Client and a Luna SA for Government HSM

1.4 Setup Synopsis

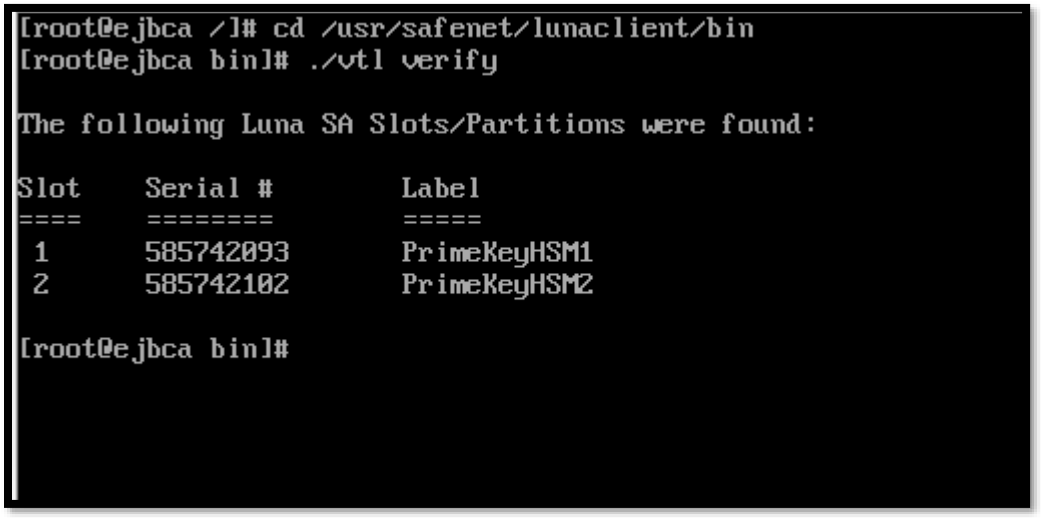
- Verify the Network Trust Link (NTL) between the EJBCA server and the HSM and that two partitions exist
- Create the Crypto Tokens and key pair for the CAs using the HSM
- Create the Certificate Profiles for the Root and Subordinate CA
- Create the Root and Subordinate CAs
- Verify the private keys for the CAs were created on the HSM

2 Verify the HSM Configuration

Verify the HSM client configuration prior to proceeding using the **`vtl verify`** command.

1. Open a terminal session and change into the Luna Client directory, typically **/usr/safenet/lunaclient/bin**
2. Enter the following command to check that the client is configured correctly and the two necessary partitions are visible. EJBCA requires distinct partitions for each CA being configured. In the case of setting up a Root CA and a Subordinate CA two, partitions must be available.

   `./vtl verify`



3 Create the Crypto Tokens and Key Pairs

EJBCA uses the concept of Crypto Tokens to manage the keys for signing, decrypting and test functions. With the SafeNet AT Luna Client installed and configured, the keys in the Crypto Token can be created and stored in the HSM for higher security.

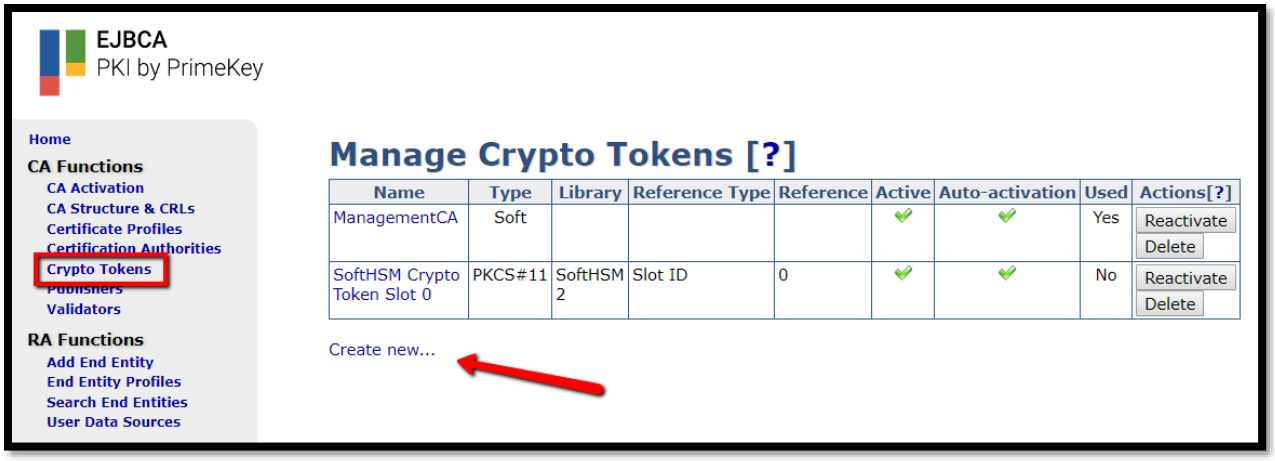
For this integration there will be two CAs used: a Root CA and a Subordinate CA that is an Issuing CA, by function. This configuration requires two Crypto Tokens be created, one for each CA, and then three key pairs be created for each Crypto Token.

3.1 Create the Root CA Crypto Token

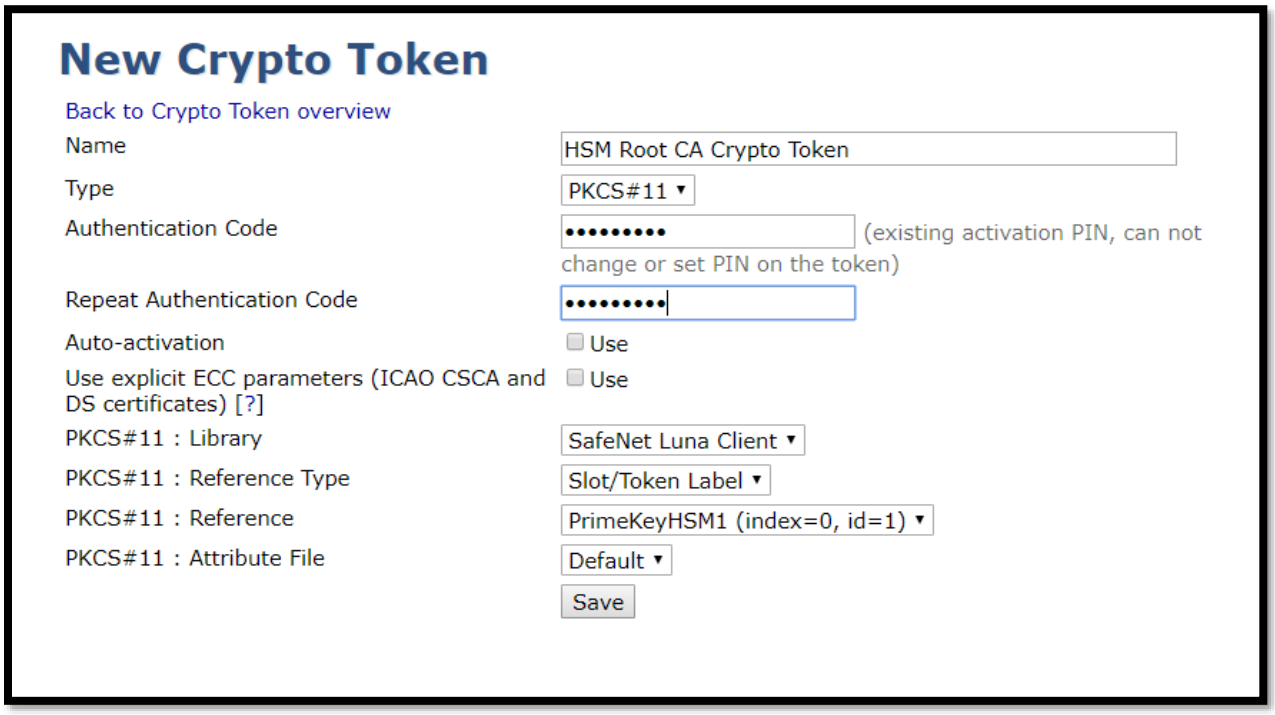
As a first step, created the Crypto Token for the Root CA. All EJBCA configuration will be done from the web interface that can be accessed via the following weblink:

`https://<Hostname or IP Address>/ejbca/adminweb`

1. Click the **Crypto Tokens** option in the **CA Functions** section, then click **Create new…**

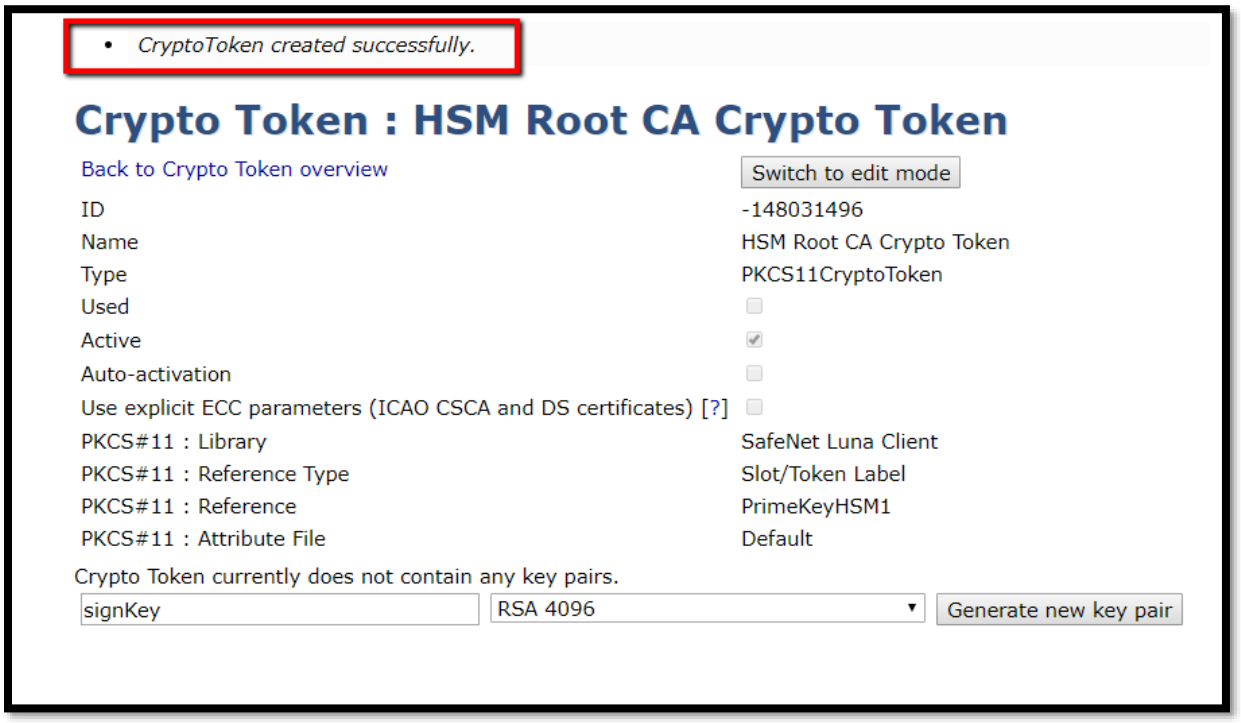


2. On the New Crypto Token page:
   - Enter a **Name** for the Crypto Token for the Root CA.
   - For **Type,** select **PKCS#11**. This will cause the **Authentication Code** fields to appear.
   - For **Library**, select **SafeNet Luna Client**. (It will appear as a drop-down option as long as the client software was installed in the default directory offered during installation.)
   - For **Reference Type**, select **Slot/Token Label**.
   - For **Reference,** select the partition to be used for the Root CA keys.
   - Enter the password for the Root CA partition in the **Authentication Code** and **Repeat Authentication Code** fields**.**



Note - auto activation is typically not enabled for a Root CA as it would be kept offline after signing certificates for Subordinate CAs.

3. **Click Save** and verify that the token was created successfully. The next step will be to generate key pairs for the token, and this will be done on this same page.

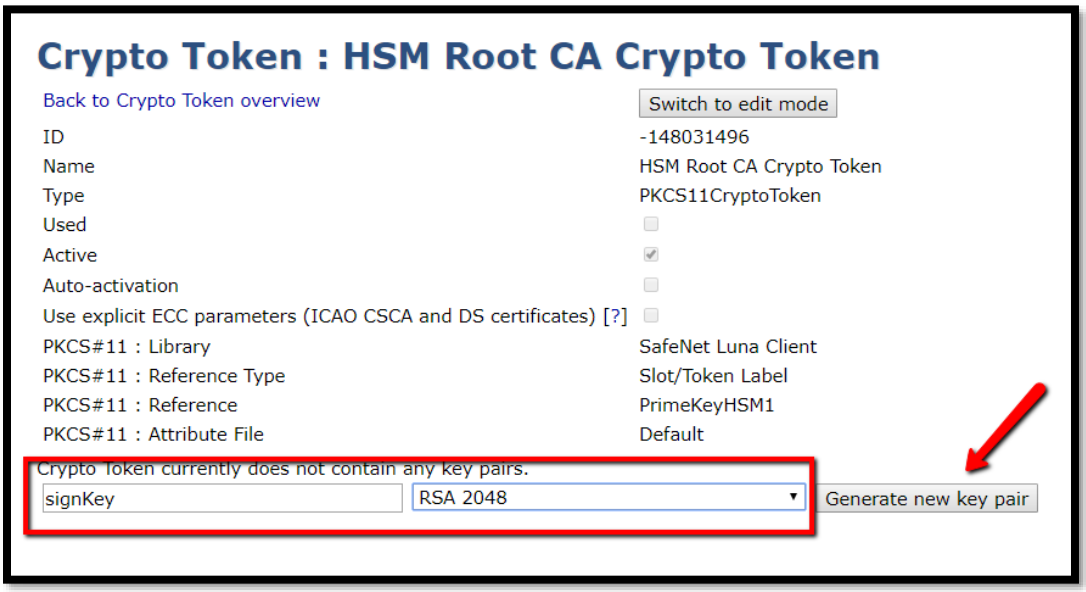


3.2 Create the Root CA Key Pairs

Three key pairs need to be created for the Crypto Token. For each pair, a name will be entered and an algorithm/bit length selected. The name is fully configurable, but by using the exact names indicated below, EJBCA will automatically know what the purpose of the key is and assign it appropriately when the CA is created.

Create the following three keys by entering the exact key name shown, selecting the algorithm/bit length, and clicking **Generate new key pair**. Repeats these actions to generate all three keys

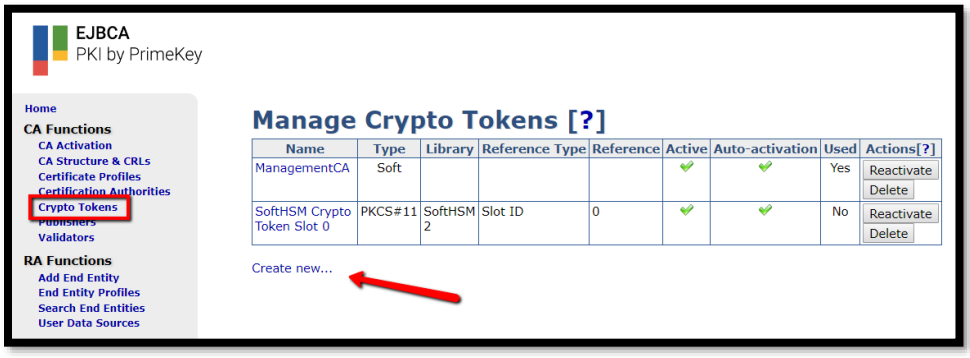
- **signKey** - used for signing certificate requests
- **defaultKey -** used for various tasks such as audit log signing
- **testKey** - smaller key (e.g. RSA 1024) used for test signings to verify and maintain HSM connectivity



3.3 Create the Subordinate CA Crypto Token

Next, create the Subordinate CA Crypto Token.

1. Click **Crypto Tokens** in the **CA Functions** section, and then click **Create new…**



2. On the New **Crypto** Token page:
   - Enter a **Name** for the crypto token (for example **HSM Issuing CA Crypto Token**).
   - For **Type,** select **PKCS#11**. This will cause the **Authentication Code** fields to appear.
   - Check the **Auto-activation** box if this CA needs to automatically reconnect to the HSM after reboot without the **Authentication Code** having to be manually entered. This would typically be done for Issuing CAs.
   - For **Library**, select **SafeNet Luna Client**.
   - For **Reference Type**, select **Slot/Token Label**.
   - For **Reference,** select the partition to be used for the Subordinate CA keys. This is a separate partition from the one used for the Root CA Crypto Token.
   - Enter the password for the partition in the **Authentication Code** and **Repeat Authentication Code** fields.

3. Click **Save** and verify that the token was created successfully.

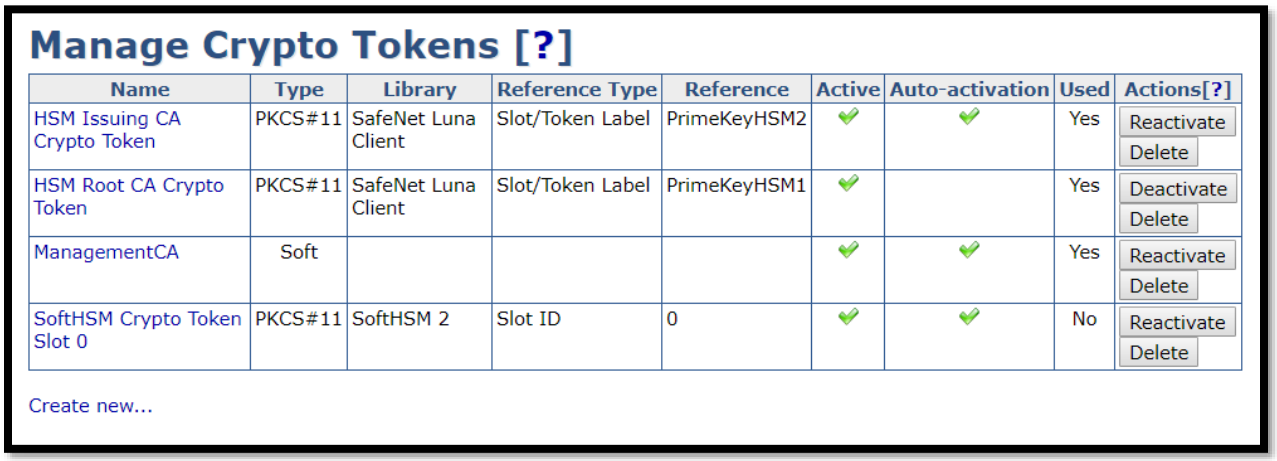
3.4 Create the Subordinate CA Key Pairs

As was done for the Root CA in Section 3.2, create the three key pairs for the Subordinate CA.

- **signKey** - used for signing certificate requests
- **defaultKey** - used for various tasks such as audit log signing
- **testKey -** a smaller key (e.g. RSA-1024) used for test signings to verify and maintain HSM connectivity

3.5 View the Crypto Tokens and Keys

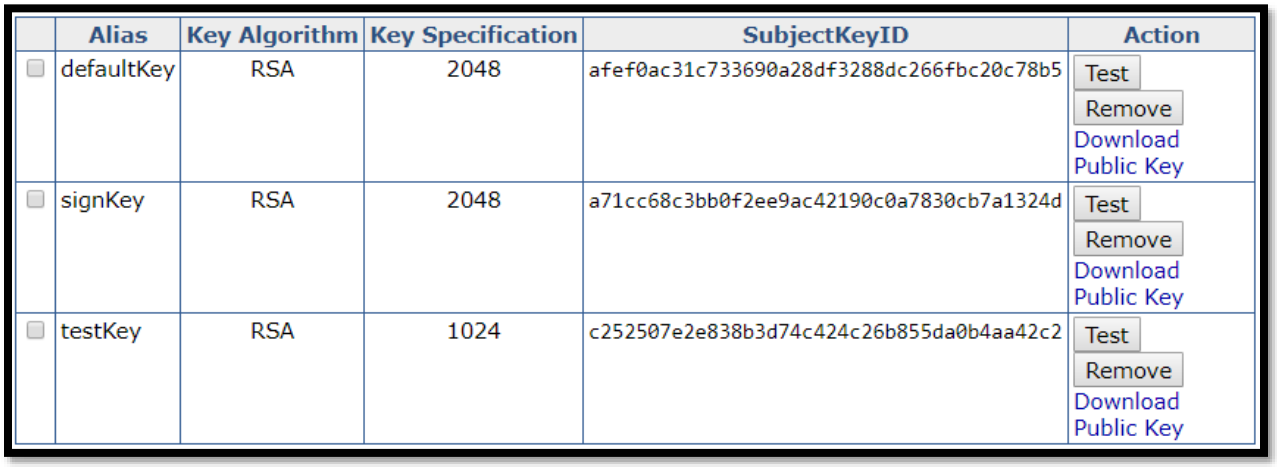
The Crypto Tokens and their keys can be viewed on the **CA Functions** -> **Crypto Tokens** page of EJBCA. Clicking on the name of a Crypto Token will show the keys created in that token.



Manage Crypto Tokens [?]

Name	Type	Library	Reference Type	Reference	Active	Auto-activation	Used	Actions[?]
HSM Issuing CA Crypto Token	PKCS#11	SafeNet Luna Client	Slot/Token Label	PrimeKeyHSM2	✔	✔	Yes	Reactivate Delete
HSM Root CA Crypto Token	PKCS#11	SafeNet Luna Client	Slot/Token Label	PrimeKeyHSM1	✔		Yes	Deactivate Delete
ManagementCA	Soft				✔	✔	Yes	Reactivate Delete
SoftHSM Crypto Token Slot 0	PKCS#11	SoftHSM 2	Slot ID	0	✔	✔	No	Reactivate Delete

Create new...



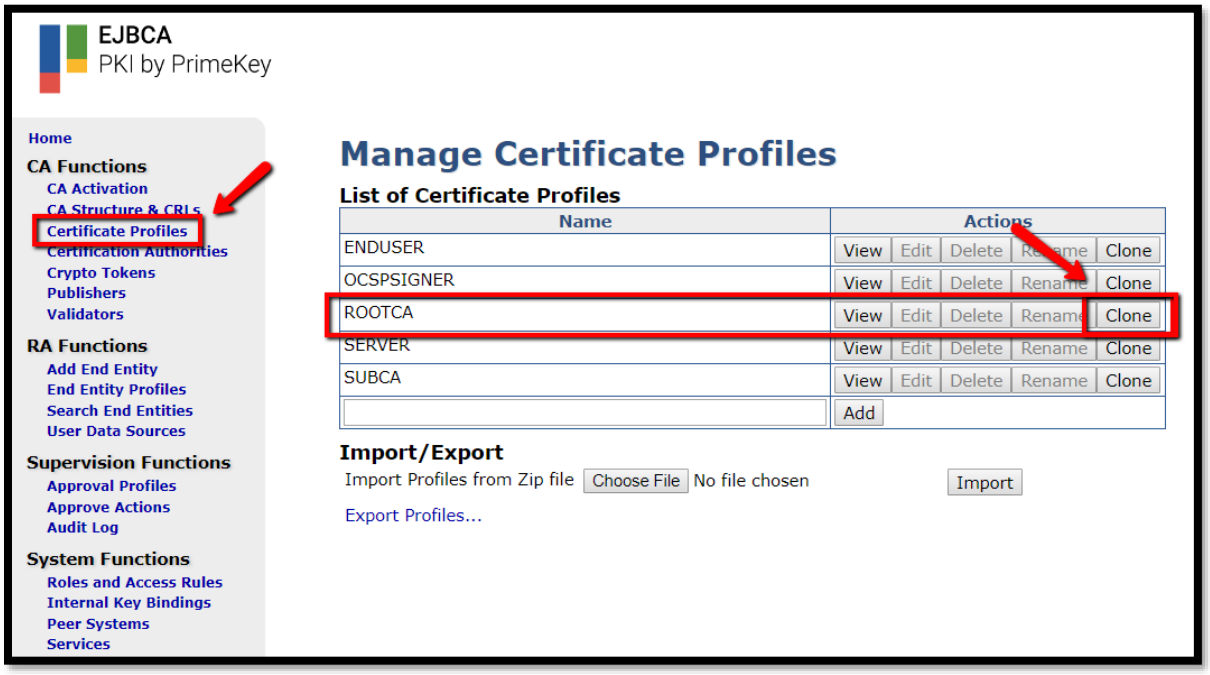
	Alias	Key Algorithm	Key Specification	SubjectKeyID	Action
☐	defaultKey	RSA	2048	afef0ac31c733690a28df3288dc266fbc20c78b5	Test Remove Download Public Key
☐	signKey	RSA	2048	a71cc68c3bb0f2ee9ac42190c0a7830cb7a1324d	Test Remove Download Public Key
☐	testKey	RSA	1024	c252507e2e838b3d74c424c26b855da0b4aa42c2	Test Remove Download Public Key

4 Create the Root and Subordinate Certification Authority

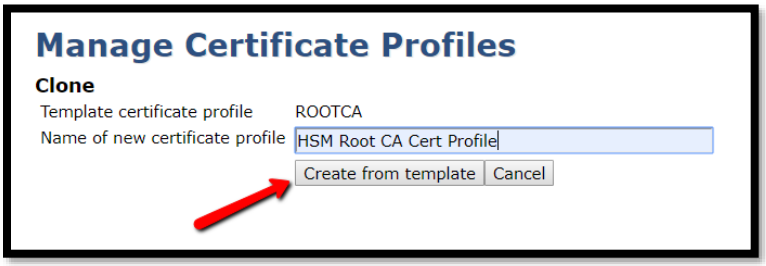
To create the Root CA and a Subordinate CA, profiles must be created for the certificates that will be used by the CAs. The following section details creating a Root and Subordinate Certificate Profile, and then using those profiles to create the Certification Authorities.

4.1 Create the Root Certificate Profile

1. From the Home page of EJBCA, click **Certificate Profiles** under **CA Functions**.
2. On the **ROOTCA** line, click **Clone**.

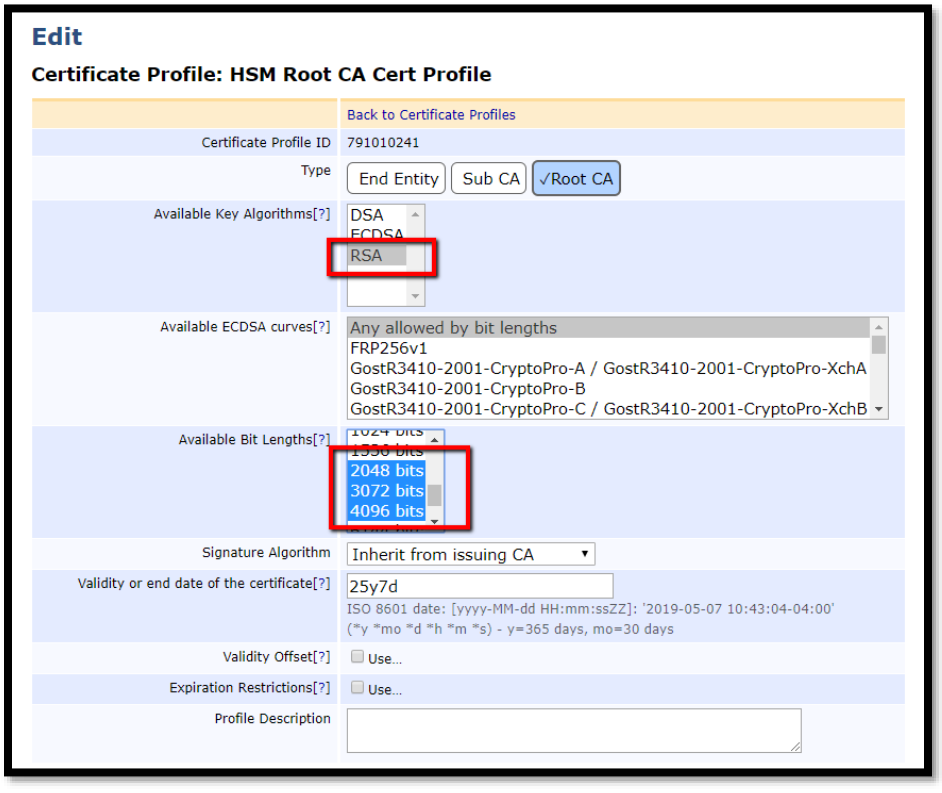


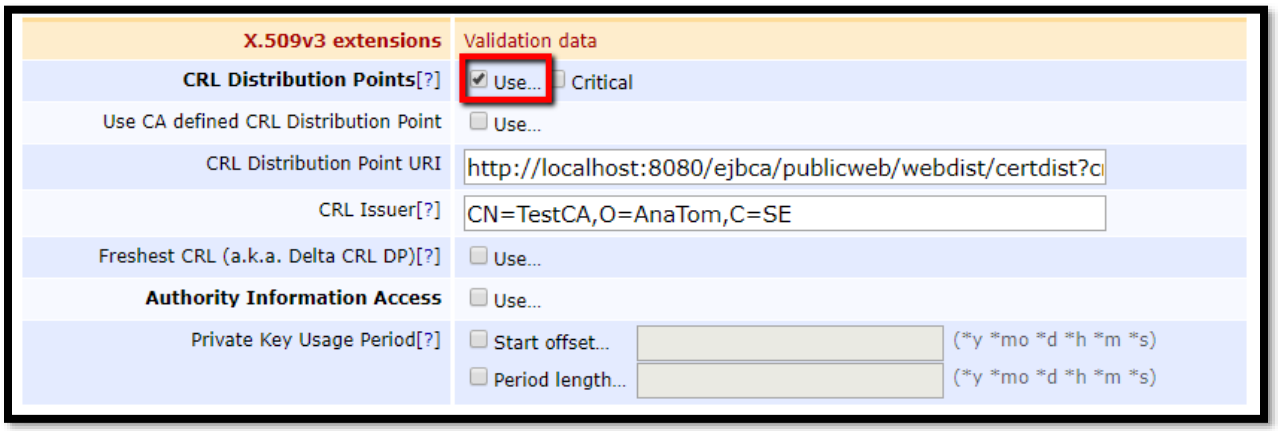
3. Enter a name for the Root CA Certificate Profile and click **Create from template**.



4. With the template created, click **Edit** to modify the settings.

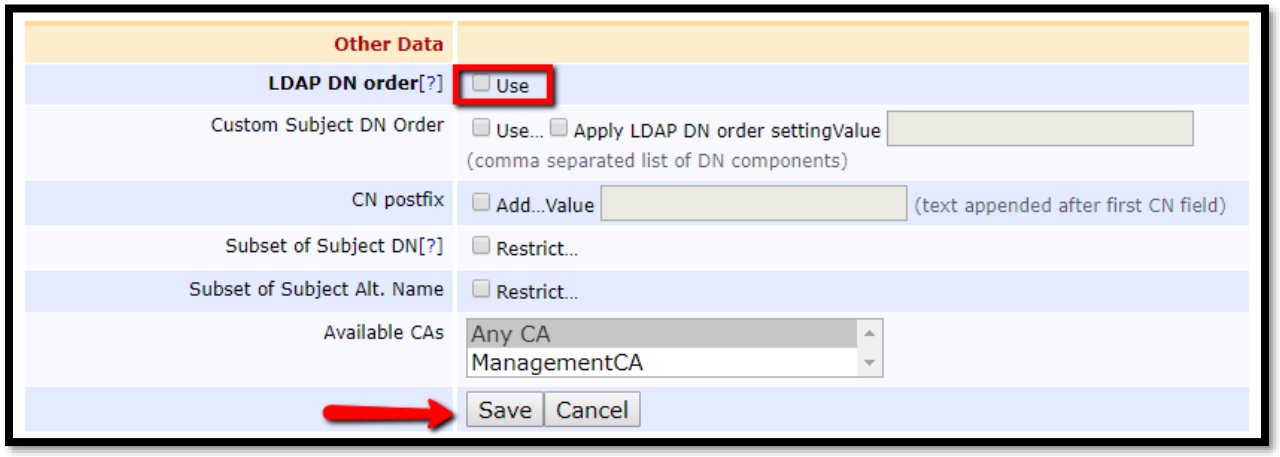
5. On the **Edit** page, set the options per your security policy and configuration. The following selections were made for this simple integration:
   - **Type** – **Root CA**
   - **Available Key Algorithms** – **RSA**
   - **Available Bit Lengths** – **2048, 3072** and **4096**
   - **CRL Distribution Points** – checked **Use** to enable publication of Certificate Revocation Lists
   - **LDAP DN Order** – cleared the **Use** checkbox in order to provide better compatibility with other systems





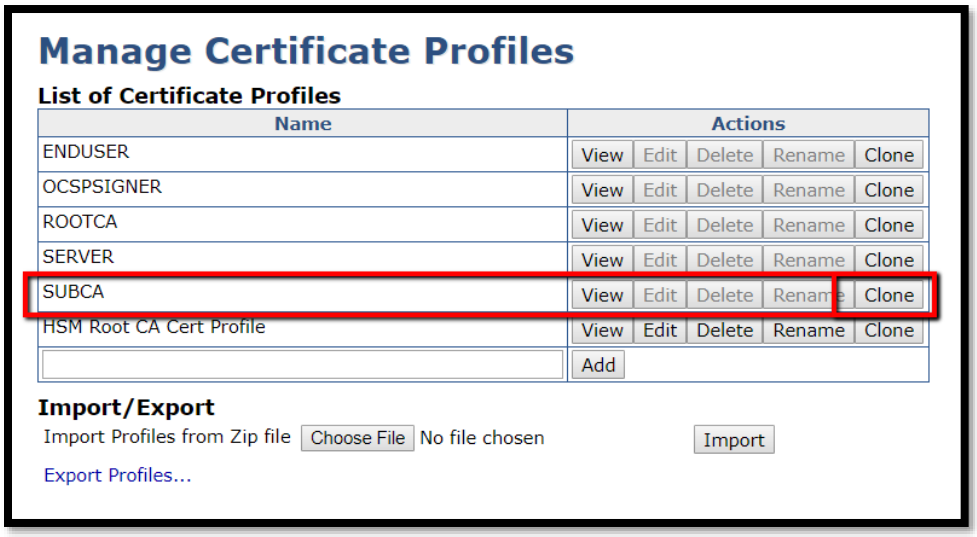
Note: In this example the CRL Distribution Point is set to "localhost" since all testing is on a single server. In production, this should be replaced with the actual hostname/URI that clients will use to retrieve the CRL.

6. Click **Save** to complete the **Root CA Certificate Profile** creation.

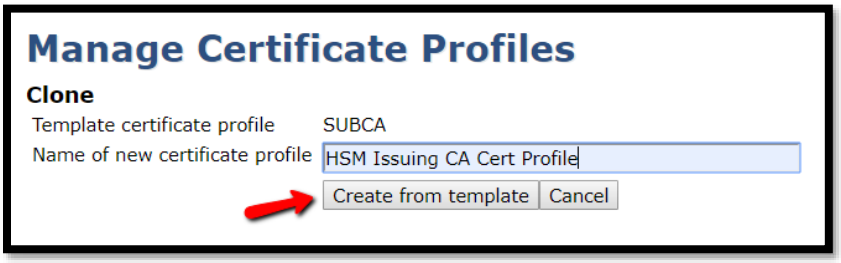


4.2 Create a Subordinate Certificate Profile

1. From the home page of EJBCA, under **CA Functions** click on **Certificate Profiles**.
2. On the **SUBCA** line click on **Clone**.

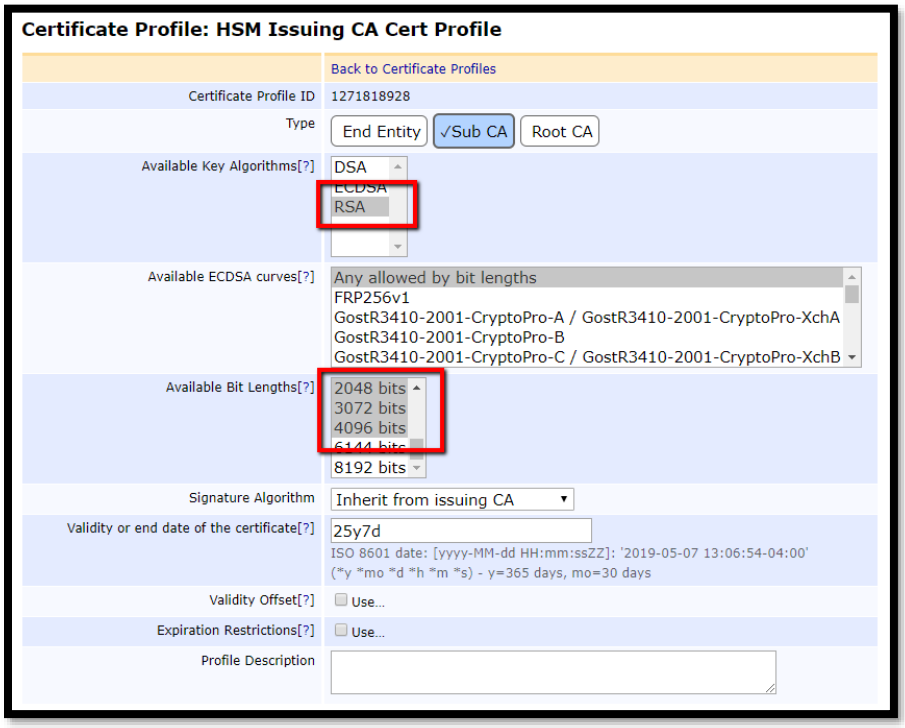


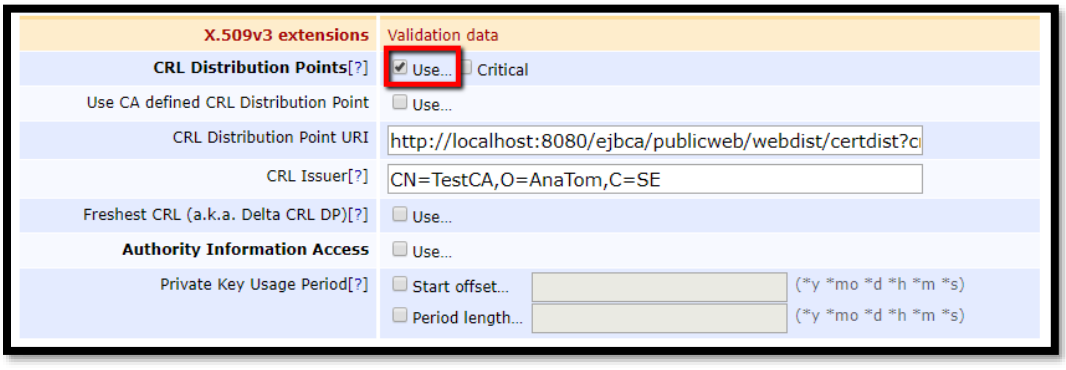
3. Enter the name for the Subordinate CA Certificate Profile and click **Create from template**.



4. With the template created, click **Edit** to modify the settings.

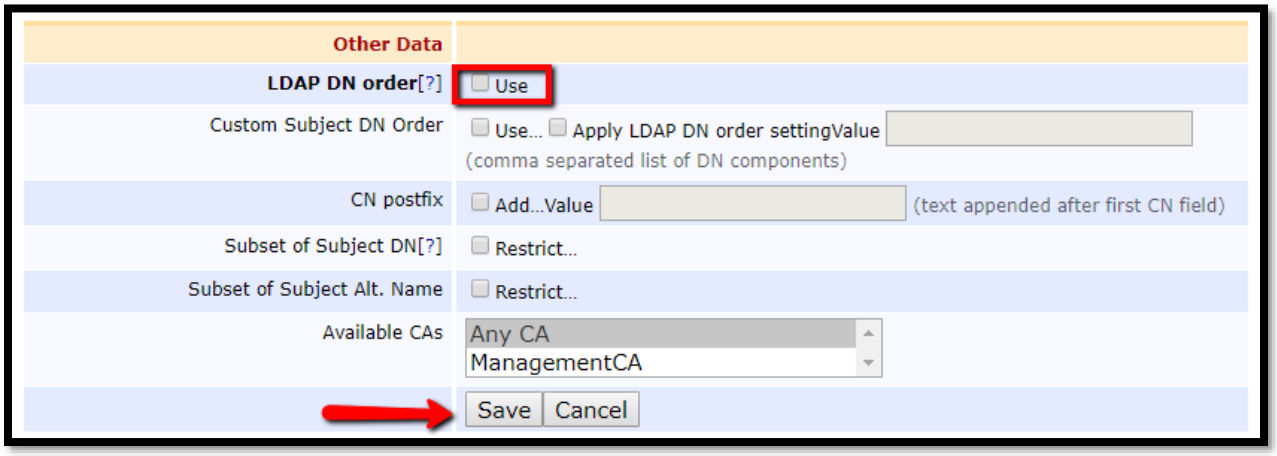
5. On the **Edit** page, set the options per your security policy and configuration. The following selections were made for this sample integration:
   - **Type** – **Sub CA**
   - **Available Key Algorithms** – **RSA**
   - **Available Bit Lengths** – **2048, 3072** and **4096**
   - **CRL Distribution Points** – checked **Use** to enable publication of Certificate Revocation Lists
   - **LDAP DN Order** – cleared the **Use** checkbox in order to provide better compatibility with other systems



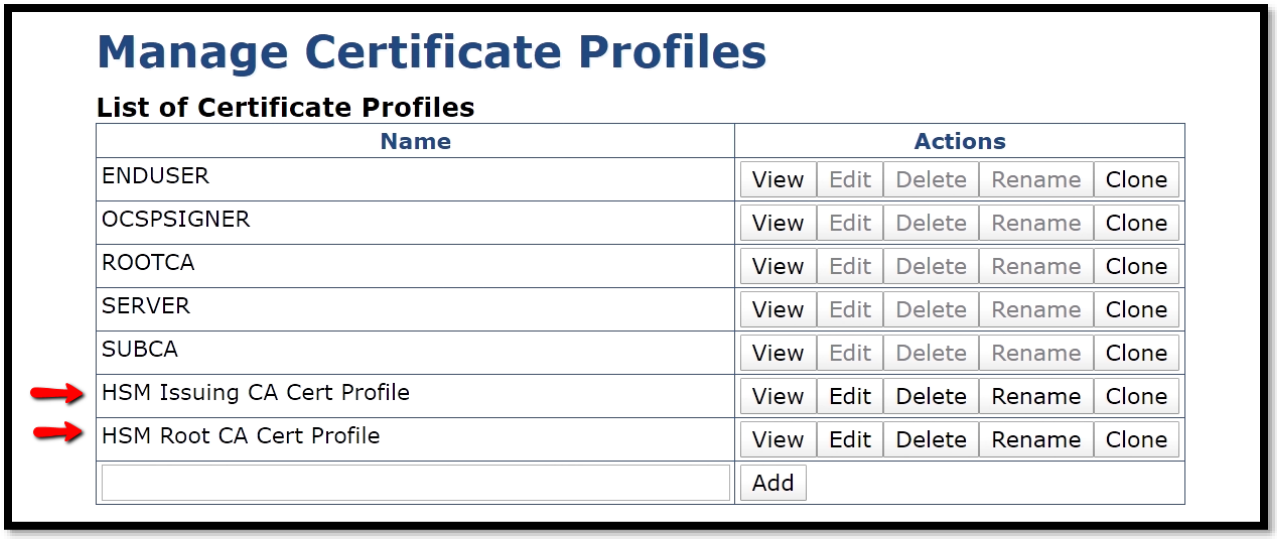


Note: In this example the CRL Distribution Point is set to "localhost" since all testing is on a single server. In production, this should be replaced with the actual hostname/URI that clients will use to retrieve the CRL.

6. Click **Save** to complete the Subordinate CA Certificate Profile creation.



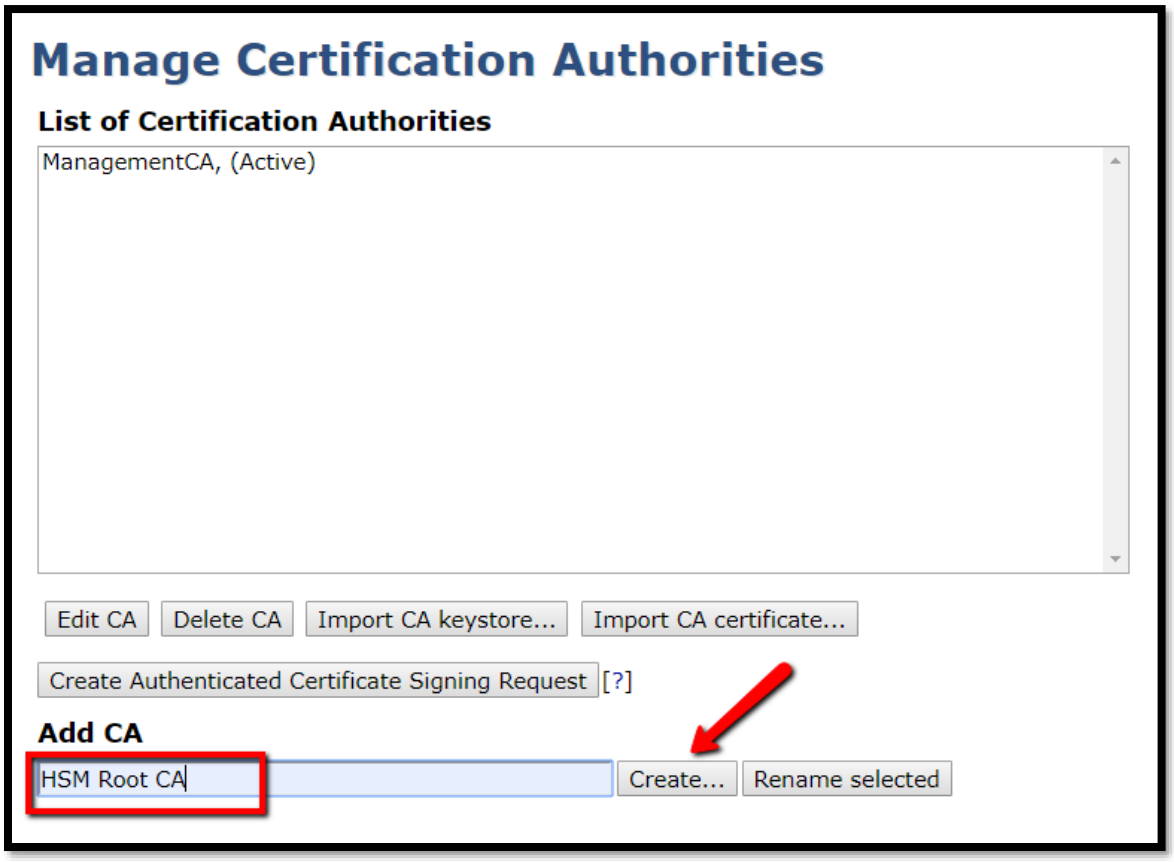
7. The two newly created Certificate Profiles will now appear in the List of Certificate Profiles.



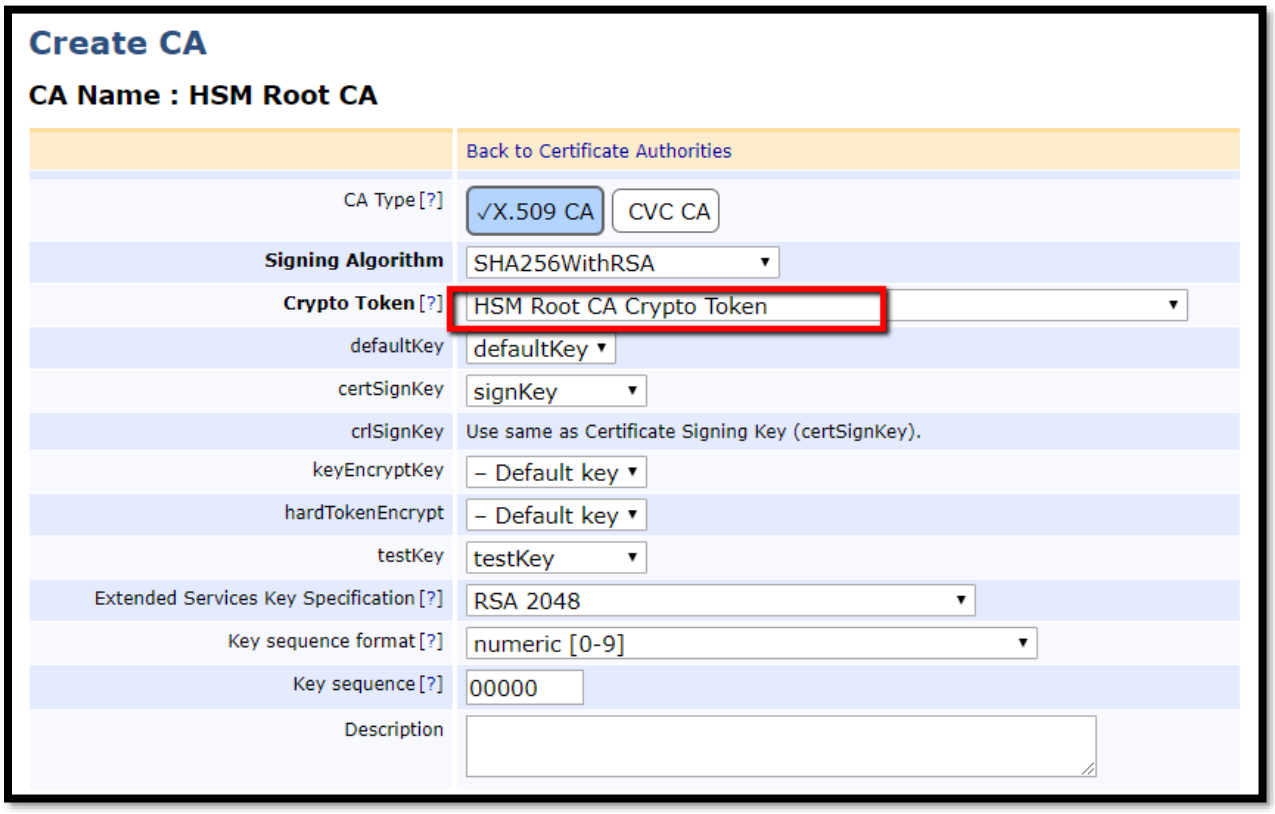
4.3 Create the Root Certification Authority

With the Certificate Profiles created, now create the CAs. Again, select options per your organization's security policy.

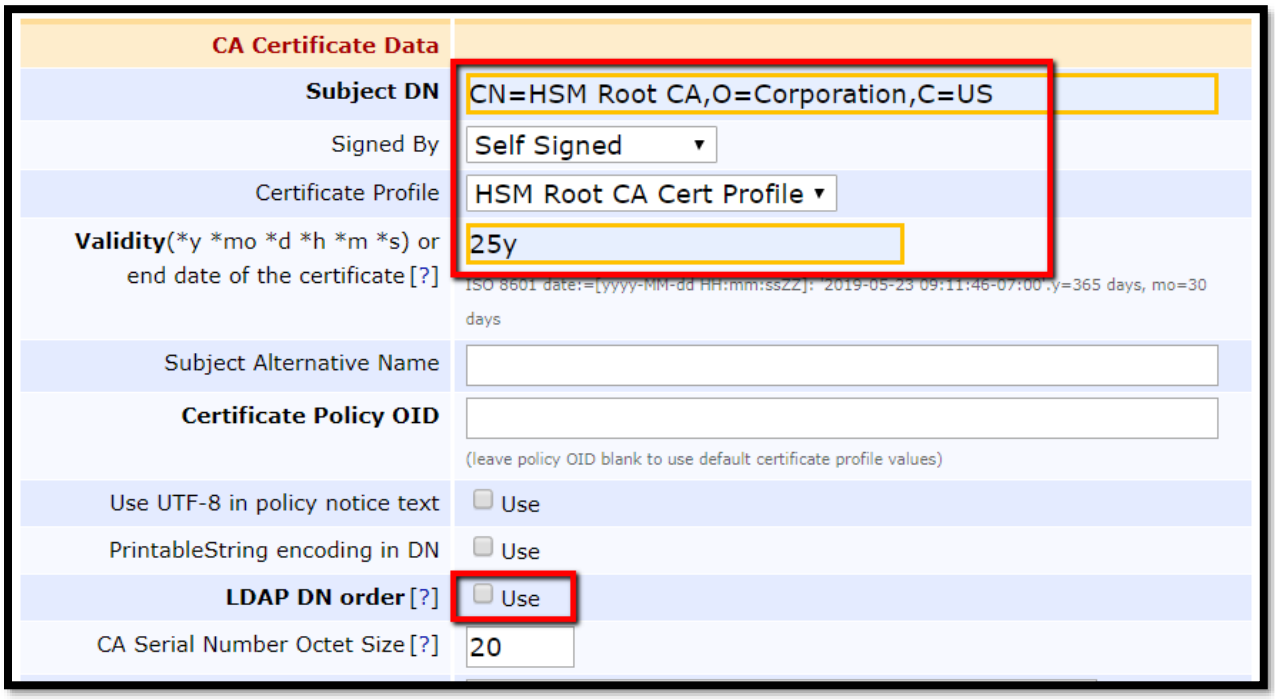
1. From the home page of EJBCA, click **Certification Authorities** in the **CA Functions** section.
2. Enter a name for the Root CA under the **Add CA** section, and then click **Create**.



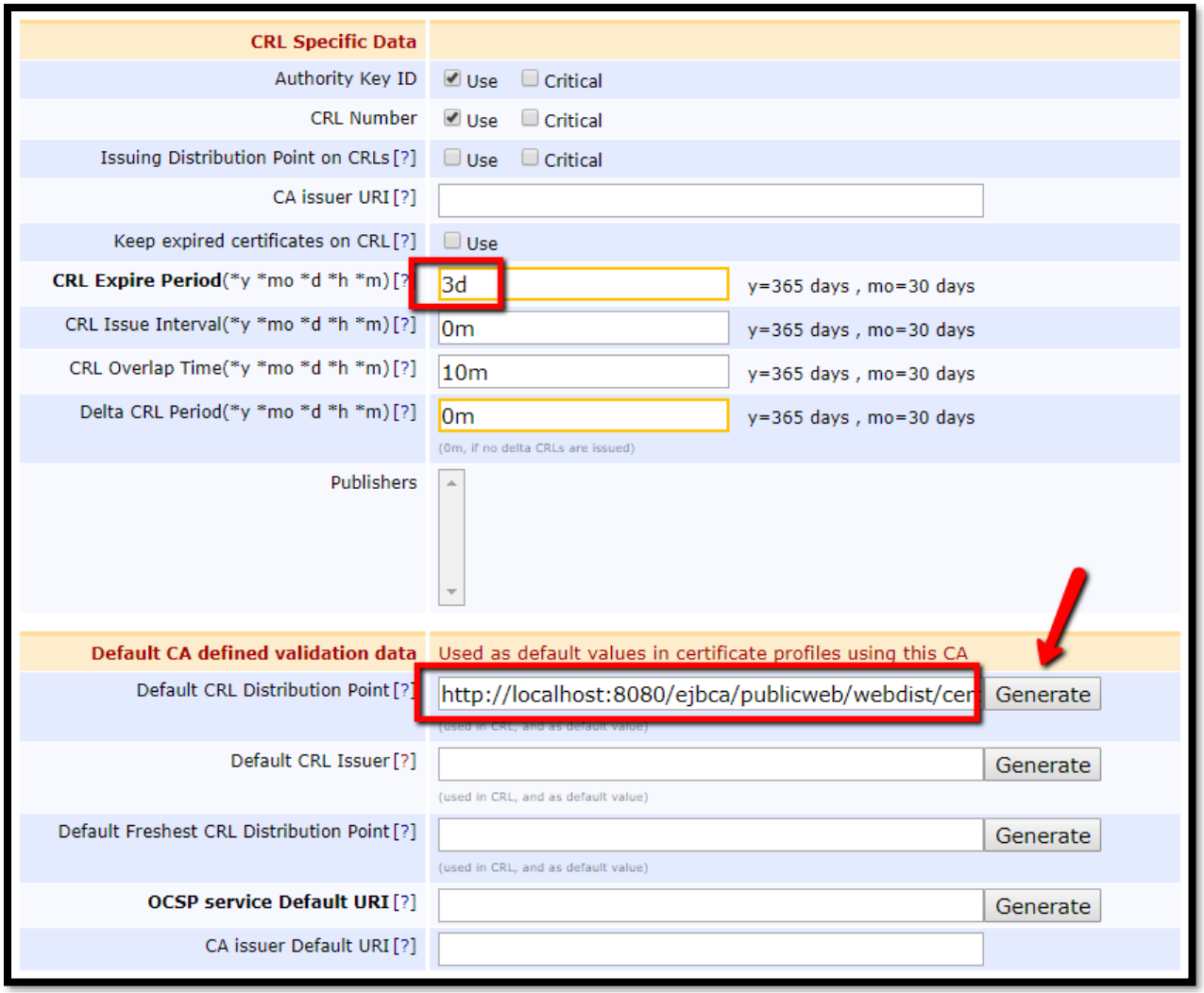
3. On the **Create CA** page, use the **Crypto Token** drop-down to select the Root CA Crypto Token created previously.
4. Verify that the various keys have been assigned correctly. The names that were used for creating the keys in this guide will be detected by EJBCA and should be assigned accordingly.



5. Under **Subject DN,** add **Organization** and **Country** values according to your configuration.
6. Leave the **Signed By** drop-down set to **Self Signed**.
7. Use the **Certificate Profile** drop-down to select the Root CA Certificate Profile previously created.
8. Set the **Validity** period for the CA certificate.
9. Uncheck the **LDAP DN Order Use** checkbox.



10. Set the **CRL Expire Period** according to your security requirements.
11. For **Default CRL Distribution Point,** either click **Generate** to have EJBCA populate the address, or enter a manual address



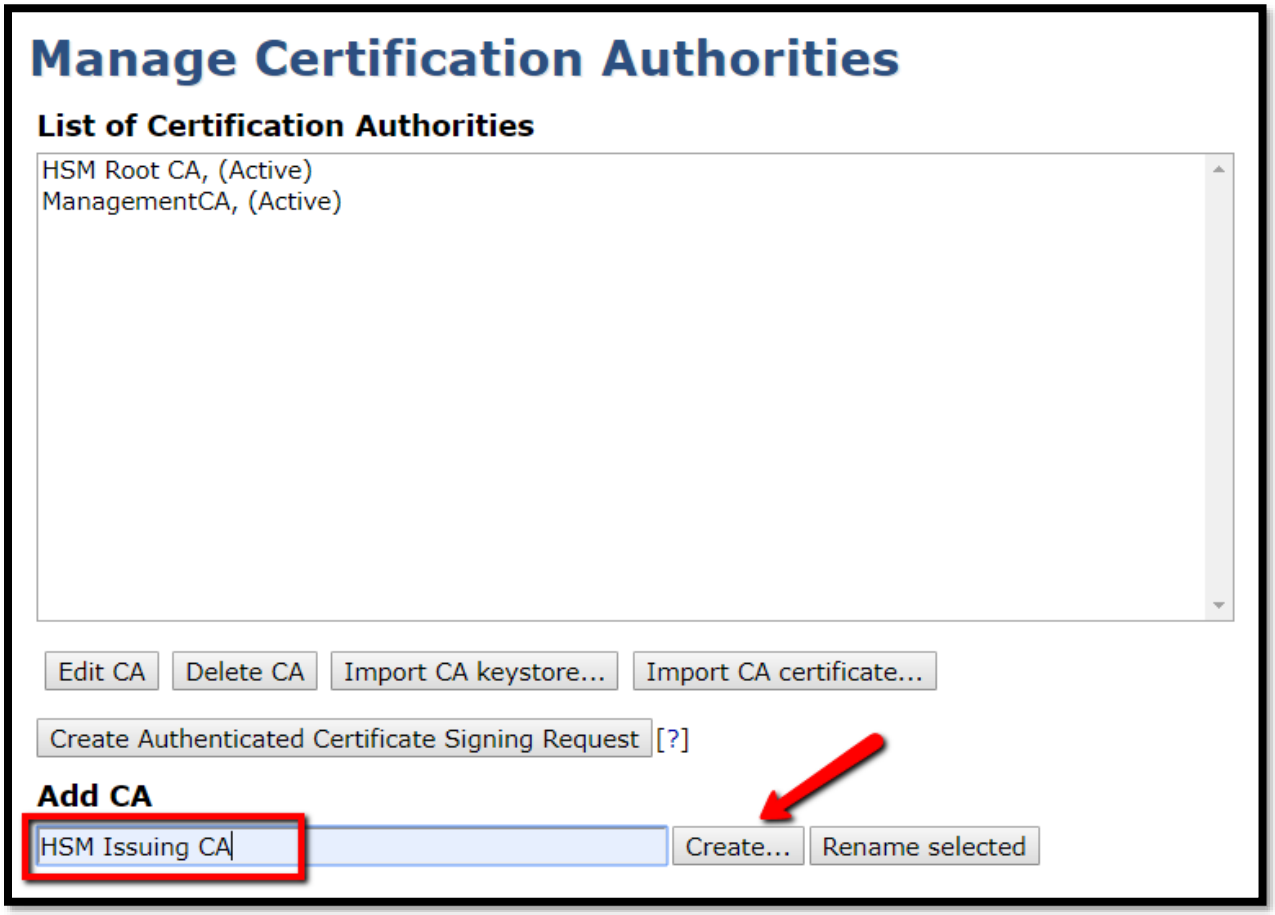
Note: In this example the CRL Distribution Point is set to "localhost" since all testing is on a single server. In production, this should be replaced with the actual hostname/URI that clients will use to retrieve the CRL.

12. With all necessary options set, click **Create**.

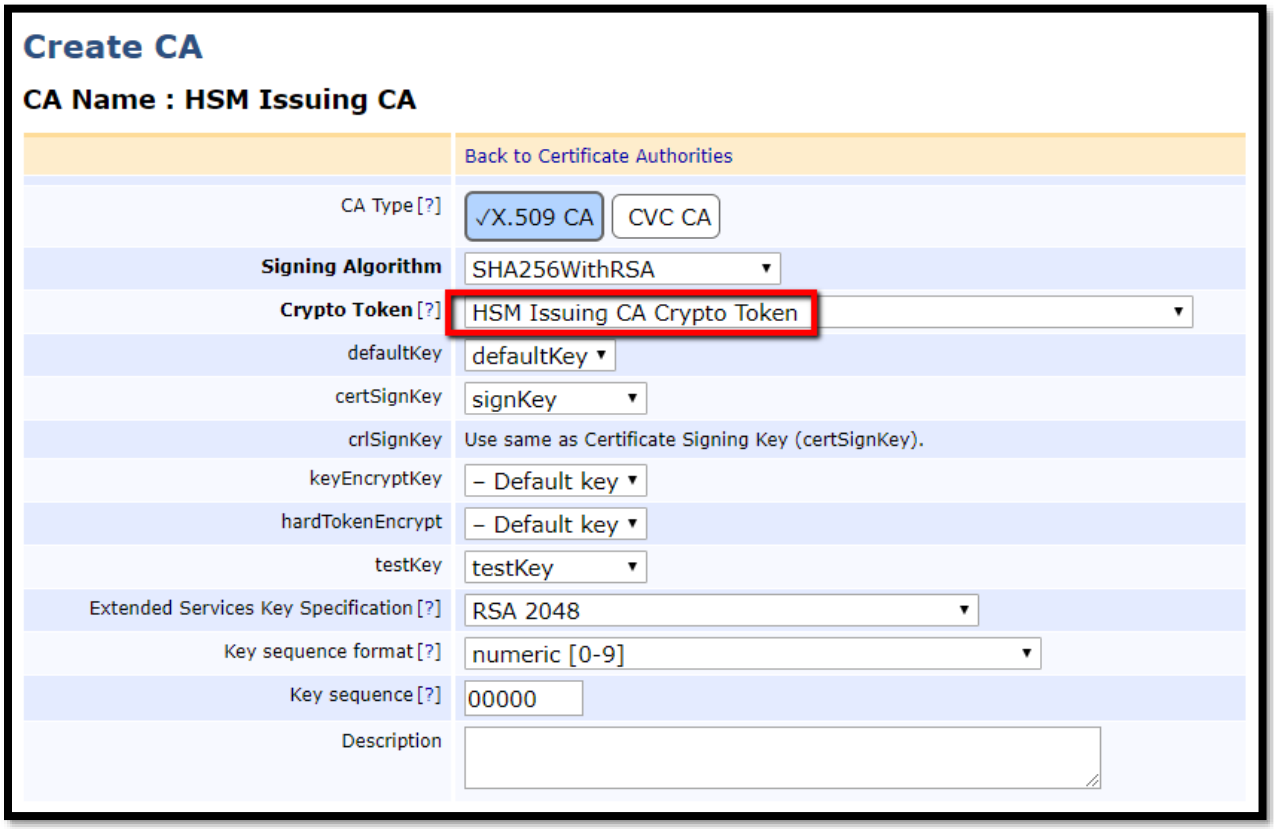
4.4 Create the Subordinate Certification Authority

Next, create the Subordinate CA(s), in this example, an Issuing CA.

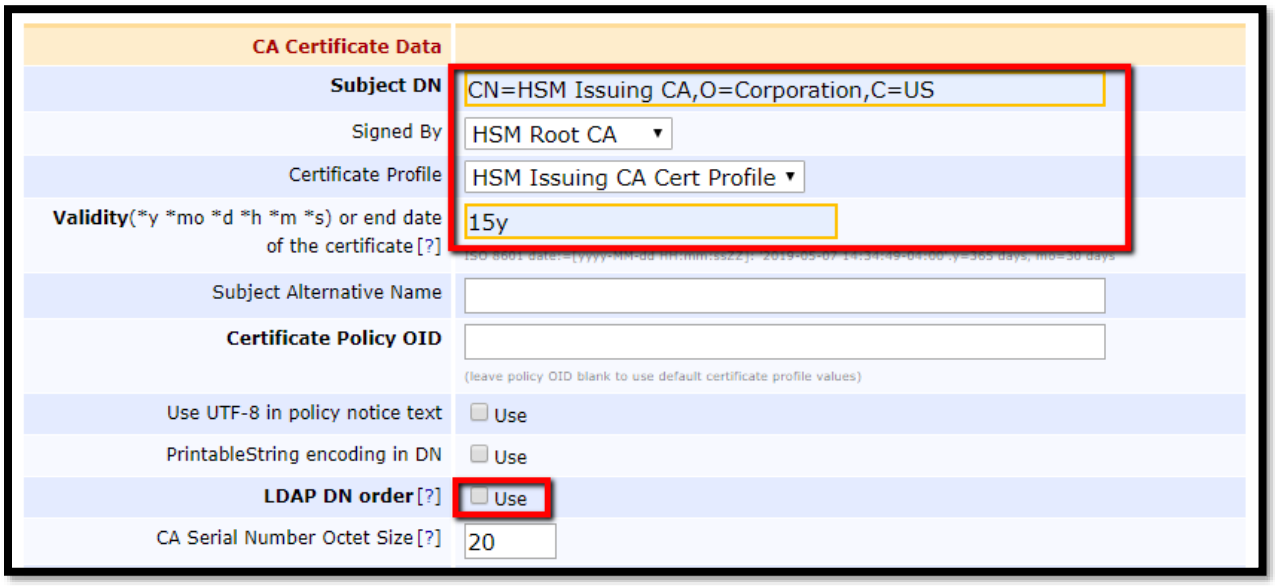
1. From the home page of EJBCA, click **Certification Authorities** under **CA Functions**.
2. Enter a name for the Subordinate CA under the **Add CA** section, and then click **Create**.



3. On the **Create CA** page, use the **Crypto Token** drop-down to select the Subordinate CA Crypto Token created previously.
4. Verify that the various keys have been assigned correctly. The names that were used for creating the keys in this guide will be detected by EJBCA and should be assigned accordingly.



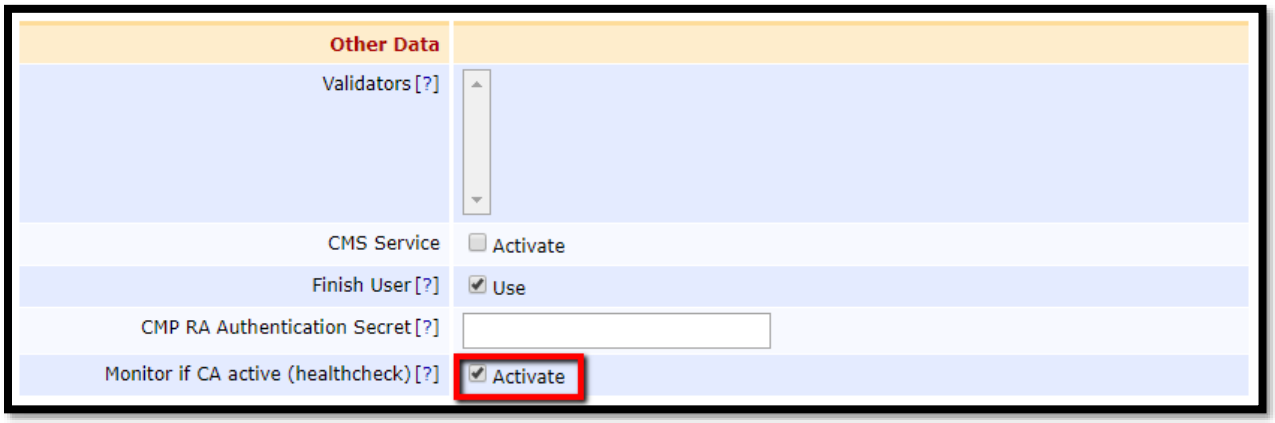
5. Under **Subject DN,** add **Organization** and **Country** values according to your configuration.
6. Use the **Signed By** drop-down to select the Root CA.
7. Use the **Certificate Profile** drop-down to select the Subordinate CA Certificate Profile previously created.
8. Set the **Validity** period for the CA certificate.
9. Uncheck the **LDAP DN Order Use** checkbox.



10. Set the **CRL Expire Period** according to your security requirements.
11. For **Default CRL Distribution Point,** either click **Generate** to have EJBCA populate the address, or enter a manual address, and then click **Create**.

Note: In this example the CRL Distribution Point is set to "localhost" since all testing is on a single server. In production, this should be replaced with the actual hostname/URI that clients will use to retrieve the CRL.

12. Because the Subordinate CA in our sample integration is an Issuing CA, **Activate** will be enabled for the **Monitor if CA active** option. Enabling this option will cause EJBCA to perform healthchecks to detect if the Issuing CA goes offline as it needs to be online and available at all times to service requests.



13. With all necessary options configured, click **Create**.

5 Verify the Keys on the HSM

The keys created in the Crypto Tokens and used by the CAs can be verified using the **cmu list** command in the Luna Client Directory on the EJBCA server.

1. Open a terminal session and change into the Luna Client directory, typically **/usr/safenet/lunaclient/bin**
2. Enter the following command to check the contents of the partitions:

   ./cmu list

3. When prompted, enter the partition to be checked and the password for the partition. The objects created by EJBCA will be listed for each partition.