Keyfactor

EJBCA

4.0.9

Integration Guide

CryptoServer HSM

SecurityServer 2.60.2 and 3.00.2

utimaco®

Imprint

Copyright 2026	Utimaco IS GmbH
	Krefelder Straße 220
	52070 Aachen
	Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226)
	EMEA +49 800-627-3081
	APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	2026-07-28
Status	PUBLISHED
Document No.	IG-2026-0088

Table of Contents

1 Introduction

The present document provides an integration guide for configuring EJBCA or Enterprise Java Bean Certification Authority with Utimaco's *CryptoServer* Hardware Security Module (HSM) over the PKCS#11 interface. During this guide two possible configuration paths for integrating an Utimaco HSM with EJBCA are described. One in which the use of an Utimaco CryptoServer HSM is configured via the web interface of EJBCA and the other one in which we integrate the HSM through editing a series of configuration files in the EJBCA home directory.

2 Overview

PrimeKey Enterprise Java Bean Certificate Authority (EJBCA) is an open-source application which is designed to manage the digital keys and certificates that make up the digital identities required to transparently automate all security-related processes in an organization. As the organization's CA system, PrimeKey EJBCA Security Manager software enables the use of digital signatures, digital receipts, encryption and permissions management services across a wide variety of applications and solutions. To enhance security of keys and certificates generated and used by a CA, EJBCA can be configured to use a Hardware Security Module (HSM). Enabling the use of a hardware security module with EJBCA not only strengthens protection of keys and certificates but also might be a necessary step towards legal conformity and certification.

CryptoServer is a hardware security module developed by Utimaco IS GmbH, i.e. a physically protected specialized computer unit designed to perform sensitive cryptographic tasks and to securely manage and store cryptographic keys and data. It can be used as a universal, independent security component for heterogeneous computer systems.

3 Requirements

You should have prepared an installed Ubuntu operating system. If you are using a PCI(e) card, you also must compile and install the necessary driver for that card.

HSM Model	CryptoServer CS-Series/S-Series/Se-Series LAN
HSM Firmware	SecurityServer 2.60.2 SecurityServer 3.00.2
Software	JBoss 5.1.0 GA EJBCA 4.0.9 Apache ant 1.8.2 MySQL Server 5 MySQL Java connector

Table 1: Software and hardware requirements

4 Components

In this section, we are going to describe how to install and configure all the components that are necessary for setting up an EJBCA. When all the components are installed successfully, we move forward to the actual EJBCA installation (section 5) and HSM configuration (section 6).

4.1 Download and Install MySQL Server

MySQL is the database system which will be used in this integration guide. To install MySQL Server, execute the following command in a terminal window.

›_ Console

```
# sudo apt-get install mysql-server
```

Once installation is done, open the MySQL command-line interface and connect as the root user.

›_ Console

```
# mysql -u root -p
Password: (enter the root password)
```

Once you logged in as the root user, now you can create the database for EJBCA. In this guide we named `ejbca` as our database name.

›_ Console

```
Mysql> CREATE database ejbca;
```

Once previous step is done successfully, a user should be created for the database, and the specific access rights should be granted to the `ejbca` database. As with the database, we give the same name `ejbca` to the newly created user.

›_ Console

```
Mysql> USE ejbca;
Mysql> CREATE user 'ejbca'@'127.0.0.1' IDENTIFIED BY 'ejbca';
Mysql> GRANT SELECT, UPDATE, DELETE, CREATE on ejbca.* to \
'ejbca'@'127.0.0.1';
```

This finishes the installation and configuration of the database system which is necessary for setting up an EJBCA.

4.2 Download and Install Apache Ant

Apache Ant can be downloaded and installed by typing below command into a terminal window. The version of Apache Ant which we used for the present integration guide is 1.8.2.

›_ Console

```
# sudo apt-get install ant
```

4.3 Download and Install the JDK

The JDK used in this guide was OpenJDK and One can install the JDK by using following command in a terminal window.

›_ Console

```
# sudo apt-get install openjdk-6-jdk
```

4.4 Download and Install the Application Server

For your reference, we used JBoss 5.1.0 GA as our application server in this guide. Download JBoss from *http://sourceforge.net/projects/jboss/files/JBoss/* and unzip the downloaded file to a directory of your choice. The path `/opt/pki/` is our installation path which we will use

throughout this guide. Let's assume that we installed the application server under `/opt/pki/jboss`. When installation is done, create a user and group named as `jboss` with a working directory `/opt/pki/jboss`.

›_ Console

```
# groupadd jboss
# useradd -d /opt/pki/jboss -g jboss jboss
# passwd jboss
```

If you use different installation path other than above path, then modify the path value whenever needed according to your path.

5 Installation of EJBCA

As mentioned, there are two possible ways in which one can configure an Utimaco HSM as a key storage/generator for an EJBCA. While in this section, a detailed installation of EJBCA application will be presented. In section 6, we will present the two different paths that can be followed in configuring the use of an Utimaco HSM with EJBCA over the PKCS#11 interface.

5.1 Download EJBCA

Download EJBCA from *http://www.ejbca.org/download.html* and unzip it to the directory of your choice. In this integration guide, we used the path `/opt/pki/ejbca` as our installation path. A common setting to EJBCA setup is the configuration of the database properties file. To edit this file, go to `/opt/pki/ejbca/conf` and edit the file named: `database.properties.sample` according to the database setup which you did in section 4.1. After you have adjusted the database properties file, rename it to `database.properties` and save the changes.

5.2 Set up the Environment Variables

To run all installed components together, a series of environment variables must be created. To do so, create a file named as a `ejbca.sh` under `/etc/profile.d` and add lines those are given below. Always remember that the paths provided in this guide are relative to your installation.

#!/bin/sh

```
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk
export JBOSS_HOME=/opt/pki/jboss
export APPSERV_HOME=/opt/pki/jboss
export ANT_OPTS=-Xmx512m
export JBOSS_HOST=0.0.0.0
export ANT_HOME=/opt/pki/ant
export EJBCA_HOME=/opt/pki/ejbca
export JBOSS_CONSOLE=/var/log/jboss.log
export JBOSSUS=jboss
export JBOSS_SERVER=default
export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$PATH
```

Once you are done editing the `ejbca.sh` file, save and close it. The file `ejbca.sh` in the `/etc/profile.d` directory will be executed whenever a bash login shell is entered. This finishes the basic configuration of all the components needed to install EJBCA.

5.3 Install EJBCA

To start EJBCA, run the below listed commands in the EJBCA home directory. Make sure that JBoss is not running when you run below commands.

›_ Console

```
# cd $EJBCA_HOME
# ant clean
# ant bootstrap
```

Once the bootstrapping process is completed, proceed with starting the JBoss application server.

›_ Console

```
# cd $JBOSS_HOME/bin
# chmod +x run.sh
# ./run.sh
```

While JBoss is running, change the current directory back to the home directory of EJBCA. After switching back to the EJBCA home directory, start the installation by running below commands.

›_ Console

```
# cd $EJBCA_HOME
# ant install
```

When the BUILD SUCCESSFUL message appears, shutdown JBoss and move to the deployment of EJBCA.

›_ Console

```
# $JBOSS_HOME/bin/shutdown.sh -s
# cd $EJBCA_HOME
# ant deploy
```

After deployment, restarting the JBoss server and EJBCA's Web Interface is now accessible through http://localhost:8080/ejbca. To be able to access EJBCA's administrator interface, one needs to import the `superadmin.p12` certificate which is generated within the p12 installation directory of EJBCA.

6 HSM Configuration

By completing section 5 successfully, you can now start setting up and deploying your CA(s). In this section, we are going to discuss the options those we have in terms of integrating EJBCA with Utimaco's CryptoServer over the PKCS#11 interface and maximizing security over the sensitive CA key material. As mentioned, there are two possible ways to integrate an Utimaco HSM into an EJBCA installation and both will be presented in the following two subsections. Regardless of which configuration one will choose, there are some steps that have to be taken in advance in order to configure a PKCS#11 slot which will then be used for the generation and storage of the CA's keys.

Use p11tool to initialize a slot on the HSM (in this example slot 1).

›_ Console

```
# ./p11tool Slot=1 InitToken=officer1
# ./p11tool Slot=1 Label=CVCA LoginSO=officer1 InitPin=user1
```

When done, use the pkcs11HSM tool from the EJBCA toolbox to generate a sign key, default key and test key on the initialized slot. The process of generating keys is listed below.

›_ Console

```
# cd $EJBCA_HOME/bin
# ./pkcs11HSM.sh generate ./libcs2_pkcs11.so 4096 signKey 1 \
PKCS11 Token \
[SunPKCS11-libcs2_pkcs11.so-slot1] Password:
# ./pkcs11HSM.sh generate ./libcs2_pkcs11.so 2048 defaultKey 1 \
PKCS11 Token \
[SunPKCS11-libcs2_pkcs11.so-slot1] Password:
# ./pkcs11HSM.sh generate ./libcs2_pkcs11.so 1024 testKey 1 \
PKCS11 Token \
[SunPKCS11-libcs2_pkcs11.so-slot1] Password:
```

Use `user1` as a password to generate above keys. This finishes the slot initialization and respective key generation upon which you can now set up a CA. It is possible to observe the pkcs11 objects created on the slot by using the p11tool like shown below.

›_ Console

```
# ./p11tool Slot=1 Login=user1 ListObjects
```

6.1 HSM integration via the EJBCA Web Interface

The most straight forward way to configure an Utimaco HSM with your CA is via the Web interface of EJBCA installation. To do so, login in to the administrator's interface as described in the EJBCA user's guide and proceed with the generation of a new CA. In the configuration menu, select the *hard Token* as the CA's token option and in the field that opens enter the properties given next.

slot 1

```
defaultKey defaultKey
certSignKey signKey
crlSignKey signKey
testKey testKey
pin user1
sharedLibrary /etc/utimaco/libcs2_pkcs11.so
```

The shared library path given previously is custom and thus caution has to be taken for your EJBCA to find the location where the library is placed. When all the fields have been filled, press the generate button and a new HSM based CA is created (Please see EJBCA user manual for details).

6.2 HSM Integration via the EJBCA Configuration Files

The alternative setup path is to directly edit the relevant configuration files in your EJBCA home directory and add the necessary information to interface of EJBCA with an Utimaco HSM via the pkcs11 API. To do so, change directory to `EJBCA/conf` directory and remove the `.sample` from the name of configuration files: `install.properties`, `catoken.properties` and `ejbca.properties`. After changing the name of configuration files, open the `install.properties` file and edit the lines given below accordingly.

```
ca.name=the CA name
ca.dn=CN=the CA name, O=organization name, C=a two letter country code
ca.tokentype=org.ejbca.core.model.ca.catoken.PKCS11CAToken
ca.tokenpassword= slot password according to your initialization
ca.tokenproperties=/opt/pki/ejbca/conf/catoken.properties
ca.keyspec=2048
ca.keytype=RSA
ca.signaturealgorithm=SHA1WithRSA
ca.validity=3650
ca.policy=null
```

Edit those properties according to the PKCS#11 slot initialization and installation path that you used in preparation of the EJBCA setup. Finally, edit the file `catoken.properties` by adding the lines given below to your configuration.

slot 1

```
defaultKey defaultKey
certSignKey signKey
crlSignKey signKey
testKey testKey
pin user1
sharedLibrary /etc/utimaco/libcs2_pkcs11.so
```

When all the files have been edited according to your custom configuration, you can start your EJBCA and use the secure environment that is offered by an HSM by following the procedure described in section 5.3.

7 Further Information

This document forms a part of the information and support which is provided by Utimaco IS GmbH. Additional documentation can be found on the product CD in the documentation directory.

All CryptoServer product documentation is also available at the Utimaco IS GmbH website: *https://utimaco.com*.

8 Contact and Support Information

You can reach us from Monday to Friday, 09.00 a.m. to 05.00 p.m., Central European Time (CET).

Utimaco IS GmbH
Krefelder Str. 220
52070 Aachen
Germany

RMA Query

If you need to send the device back to Utimaco IS GmbH, please open a new RMA case (Return Merchandise Authorization). We request that you use the following web address. RMA cases cannot be opened by email or phone.

https://support.hsm.utimaco.com/support/rma/new

Other Support Queries

- Mail (preferred contact method)
  support@utimaco.com
  Attach the diagnostic information to your email.
- Web portal
  https://support.hsm.utimaco.com/support/cases/new/
  The diagnostic information will be requested in our response if necessary.
- By phone
  AMERICAS +1-844-UTIMACO (+1 844-884-6226)
  EMEA +49 800-627-3081
  APAC +81 800-919-1301
  The diagnostic information will be requested in our response if necessary.