OpenXPKI

3.26.0

Integration Guide

CryptoServer HSM

SecurityServer 4.55.0.0

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	2026-07-20
Status	PUBLISHED
Document No.	IG-2026-0073

Table of Contents

1 Introduction

This guide is part of the information and support provided by Utimaco. Additional documentation produced to support your Utimaco SecurityServer product can be found in the document directory of the Utimaco SecurityServer product bundle.

All Utimaco SecurityServer product documentation is available from Utimaco's website at *https://utimaco.com/*.

1.1 About This Guide

This guide provides an integration guide explaining how to integrate an Utimaco CryptoServer Hardware Security Module (HSM) with OpenXPKI. Utimaco securely generates and stores the private keys of Root CA and Issuing CA used by OpenXPKI.

1.2 Target Audience for This Guide

This guide is intended for administrators of OpenXPKI Administrator and of Utimaco HSMs.

1.3 Document Conventions

The following conventions are used in this guide:

Convention	Use	Example
Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Select **Details** and click on **Properties** button
`Monospaced`	Code that is given for explanation or as an example, file paths	`certreq.exe -new request.inf IISCertRequest.csr`
Italic	References and important terms	Operating system listed in *Tested Versions*

Table 1: Document conventions

We use special icons to highlight the most important notes and information.

Here you will find important safety information that should be followed.

Here you will find additional notes or supplementary information.

This message marks the result expected after the successful execution of an instruction.

1.4 Abbreviations

The following abbreviations are used in this guide:

Abbreviation	Meaning
CA	Certificate Authority
CRL	Certificate Revocation List
CSADM	CryptoServer Command-line Administration Tool
CSR	Certificate Signing Request
FCGI	Fast Common Gateway Interface
GPG	GNU Privacy Guard
GUI	Graphical User Interface

Abbreviation	Meaning
HSM	Hardware Security Module
LAN	Local Area Network
MBK	Master Backup Key
OS	Operating System
PCIe	PCI Express Interface
PIN	Personal Identification number
PKI	Public Key Infrastructure
PKCS#11	Public-Key Cryptography Standard #11
RA	Registration Authority
RDBMS	Relational Database Management Systems
RSA	Rivest, Shamir, Adleman (cryptosystem)
SCEP	Simple Certificate Enrollment Protocol
TLS	Transport Layer Security
UI	User Interface

Abbreviation	Meaning
URL	Uniform Resource Locator

Table 2: List of abbreviations

2 Overview

2.1 OpenXPKI

OpenXPKI is an easy-to-deploy and easy-to-use RA/CA software that makes handling certificates easy. OpenXPKI aims to be an enterprise-scale Public Key Infrastructure (PKI) solution, supporting well established infrastructure components like RDBMS and Hardware Security Modules (HSMs). It started as the successor of OpenCA and builds on the experience gained while developing it as well as on our experience in large public key infrastructures.

2.2 Utimaco SecurityServer HSM

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

2.3 Utimaco u.trust Anchor

u.trust Anchor is the next generation hardware security module platform developed by Utimaco IS GmbH. u.trust Anchor is a physically protected specialized computer unit designed for true multi-tenancy and to perform sensitive cryptographic tasks and to securely manage as well as store cryptographic keys and data. It can be used as a universal, independent security component for heterogeneous computer systems.

3 Integration Requirements and Prerequisites

Ensure the system environment you will be using meets the following hardware and software requirements.

3.1 Tested Versions

The integrations that have been successfully tested with the Utimaco HSM with OpenXPKI.

Operating System	OpenXPKI Version	Utimaco Security Server Version	Utimaco HSM
Debian 12	3.26.0	SecurityServer 4.55.0.0	CryptoServer CSe-Series/Se-Series u.trust Anchor Se*k and u.trust Anchor CSAR

Table 3: List of tested versions

3.2 Software Requirements

Software	Software Requirements
HSM Utility	PKCS#11 Tool Version 2 (p11tool2)
HSM Interfaces	SecurityServer PKCS#11 Provider

Table 4: List of software requirements

3.3 Hardware Requirements

Hardware	Hardware Requirements
Utimaco LAN HSM	CryptoServer CSe-Series/Se-Series LAN with firmware SecurityServer 4.55.0.0 or higher u.trust Anchor Se*k and CSAR Series LAN with firmware 4.55.0.0 or higher
Utimaco PCI-e HSM	CryptoServer CSe-Series/Se-Series PCI-e with firmware SecurityServer 4.55.0.0 or higher u.trust Anchor Se*k and CSAR Series PCI-e with firmware 4.55.0.0 or higher

Table 5: List of hardware requirements

Set up an account on the Utimaco support portal and request download access at the following URL https://support.hsm.utimaco.com/.

3.4 3.4 Prerequisites

Before you begin, please ensure that:

- The Utimaco CryptoServer HSM is set up and configured. Refer to the CryptoServer documentation to set up the HSM.
- The MBK has been created and stored onto each HSM. Refer to the CryptoServer documentation to set up the MBK.
- The CryptoServer Default Admin has been replaced with a new admin user.
- The operating system used is listed in *Tested Versions*.
- The SecurityServer is listed in *Tested Versions*.
- The PKCS#11 library is set up and configured as per your environment.
- You refer to the CryptoServer documentation to set up and configure the PKCS#11 library.

- You have a user with root privileges as it is required to install the packages.
- You allow port 8080 through the firewall.

4 Installing and Configuring Utimaco SecurityServer Software

4.1 Download and Install Utimaco Software

If you have not already done so, please create and request a Utimaco Support Portal Account. This will allow you to download the software components needed for this installation.

1. Copy the downloaded software at the appropriate location on the OpenXPKI server.
2. Create a utimaco folder under `/opt` directory and further create 2 directories `/opt/utimaco/bin` and `/opt/utimaco/lib`.

›_ Console

```
# mkdir -p /opt/utimaco/bin

# mkdir /opt/utimaco/lib
```

3. Copy the pkcs11 library file libcs_pkcs11_R3.so from Utimaco SecurityServer software to the `/opt/utimaco/lib` directory.

›_ Console

```
# cp ~/path_to_application_folder/lib/libcs_pkcs11_R3.so /opt/utimaco/lib
```

4. Copy the csadm and p11tool2 files from Utimaco SecurityServer software to `/opt/utimaco/bin` directory and make both the files executable.

›_ Console

```
# cd ~/path_to_application_folder

# cp csadm p11tool2 /opt/utimaco/bin

# chmod +x /opt/utimaco/bin/csadm /opt/utimaco/bin/p11tool2
```

4.2 SecurityServer PKCS#11 Configuration

1. Create the directory `/etc/utimaco`. Locate the Utimaco PKCS#11 configuration file in your SecurityServer directory, `Linux/x86-64/Crypto_APIS/PKCS11_R3/sample`.

Copy the Utimaco PKCS#11 configuration file `cs_pkcs11_R3.cfg` into `/etc/utimaco` directory.

›_ Console

```
# mkdir /etc/utimaco

# cd <install directory>/Software/Linux/x86-64/Crypto_APIs/PKCS11_R3/sample # cp
cs_pkcs11_R3.cfg /etc/utimaco # cd /etc/utimaco
```

2. Edit the `cs_pkcs11_R3.cfg` file and make the appropriate changes to the file.

›cs_pkcs11_R3.cfg

```
[Global]
# For unix:
Logpath = /tmp
# Loglevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)
Logging = 1
Keepalive = true
# Set the Device to connect with
[CryptoServer]
# Device specifier
Device = <HSM_IP>
```

For more information regarding the commands and command parameters please check the Utimaco CryptoServer documentation. The device may be a CryptoServer (PCIe or LAN) device. The device line will follow one of these patterns, based on the HSM form-factor: Device = 288@<HSM IP address> Hardware (LAN) HSM

OR

Device = /dev/cs2.0 Hardware (PCIe) HSM

To make your testing easier, it would be good to enable the PKCS#11 log file. That can be enabled by editing the **Logging Loglevel**. Set the **LogPath** and **Logging Loglevel** to 1. For testing you may want to increase it to 4.

The added **LogPath** points to a writable directory, not to a file.

If you encounter problems, check the log file named `cs_pkcs11_R3.log` in the **LogPath** defined directory. When you are done testing, you should change **Logging** to 1 or 2. This will limit the logging to only critical and important messages.

4.3 Create SO User and Initialize a Slot

You must initialize a slot with a custom label using p11tool2.

First using p11tool2, create the SO or Security Officer and using the p11tool2 command initialize the slot that you want to use, as well as the slot user, as shown below.

›_ Console

```
# p11tool2 slot=<slot_no> Label=<token_label> Login=ADMIN,ADMIN.key
InitToken=ask

# p11tool2 slot=<slot_no> LoginSO=ask InitPin=ask
```

Figure 1 : Slot Initialization output

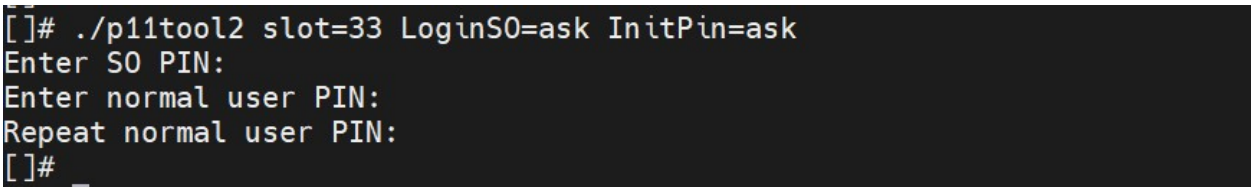

Figure 2 : Slot Initialization output

5 Integrating OpenXPKI with Utimaco HSM

5.1 Installing Libp11

1. Make sure that openssl version 3.x is installed. If it is not, then install openssl 3.x from *https://www.openssl.org/source/*.

›_ Console

```
# openssl version -a
```

```
root@sandip-openxpkinew:~/libp11-0.4.12# openssl version -a
OpenSSL 3.0.9 30 May 2023 (Library: OpenSSL 3.0.9 30 May 2023)
built on: Tue May 30 16:12:36 2023 UTC
platform: debian-amd64
options:  bn(64,64)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -fzero-call-used-regs=used-gpr -DOPENSSL_TLS_SECURITY_LEVEL=2 -Wa,--
noexecstack -g -O2 -ffile-prefix-map=/build/openssl-HFEiK2/openssl-3.0.9=. -fstack-protector-strong -Wformat -Werror=format-se
curity -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDEBUG -Wdate-time -D_FORTIFY_SOURCE=2
OPENSSLDIR: "/usr/lib/ssl"
ENGINESDIR: "/usr/lib/x86_64-linux-gnu/engines-3"
MODULESDIR: "/usr/lib/x86_64-linux-gnu/ossl-modules"
Seeding source: os-specific
CPUINFO: OPENSSL_ia32cap=0xfeda32034f8bffff:0x800d19e2fb9
root@sandip-openxpkinew:~/libp11-0.4.12#
```

Figure 3 : Openssl version

2. Download the latest libp11 package from *Releases · OpenSC/libp11 · GitHub*.

›_ Console

```
# wget https://github.com/OpenSC/libp11/releases/download/libp110.4.12/
libp11-0.4.12.tar.gz
```

3. Extract the compressed file.

›_ Console

```
# tar -xvf libp11-0.4.12.tar.gz
```

4. Go to libp11 directory, build, and install libp11using the following commands.

›_ Console

```
# cd libp11-0.4.12

# ./configure prefix="/usr/local/libp11/"

# make

# make install

# export LD_LIBRARY_PATH= /usr/local/libp11/lib/:$LD_LIBRARY_PATH
```

```
root@openxpki:~/libp11-0.4.12# ./configure prefix="/usr/local/libp11/"
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking target system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether make supports nested variables... (cached) yes
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
```

Figure 4 : Libp11 installation

```
checking for library containing dlopen... none required
checking whether gcc is Clang... no
checking whether pthreads work with "-pthread" and "-lpthread"... yes
checking for joinable pthread attribute... PTHREAD_CREATE_JOINABLE
checking whether more special flags are required for pthreads... no
checking for PTHREAD_PRIO_INHERIT... yes
checking for libcrypto >= 0.9.8... yes
checking if libtool needs -no-undefined flag to build shared libraries... no
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating src/Makefile
config.status: creating src/libp11.pc
config.status: creating src/libp11.rc
config.status: creating src/pkcs11.rc
config.status: creating doc/Makefile
config.status: creating doc/doxygen.conf
config.status: creating examples/Makefile
config.status: creating tests/Makefile
config.status: creating src/config.h
config.status: executing depfiles commands
config.status: executing libtool commands
configure: creating src/libp11.map

libp11 has been configured with the following options:

Version:                0.4.12
libp11 directory:       /usr/local/libp11/lib
Engine directory:       /usr/lib/x86_64-linux-gnu/engines-3
Default PKCS11 module:
API doc support:        no

Host:                   x86_64-pc-linux-gnu
Compiler:               gcc
Preprocessor flags:
Compiler flags:         -g -O2 -pthread
Linker flags:
Libraries:              -lpthread

OPENSSL_CFLAGS:
OPENSSL_LIBS:           -lcrypto
```

Figure 5 : Libp11 installation

5.2 Configuring OpenSSL to use Utimaco SecurityServer

5.2.1 Setting up Utimaco SecurityServer Library in OpenSSL Configuration File

1. Open the file `/etc/pki/tls/openssl.cnf` and enter the following line in the first line of the file.

openssl.cnf

```
openssl_conf = openssl_init
```

2. Enter the following lines at the end of the `openssl.cnf` file.

openssl.cnf

```
[openssl_init]
engines=engine_section
[engine_section]
pkcs11 = pkcs11_section
[pkcs11_section]
engine_id = pkcs11
dynamic_path = /usr/lib/x86_64-linux-gnu/engines-3/pkcs11.so
MODULE_PATH = /opt/utimaco/lib/libcs_pkcs11_R3.so
init = 0
```

3. Below is complete sample `openssl.cnf` file.

openssl.cnf

```
openssl_conf = openssl_init
HOME = .
config_diagnostics = 1
oid_section = new_oids
[ new_oids ]
tsa_policy1 = 1.2.3.4.1
tsa_policy2 = 1.2.3.4.5.6
tsa_policy3 = 1.2.3.4.5.7
[openssl_init]
providers = provider_sect
[provider_sect]
default = default_sect
[default_sect]
[ ca ]
default_ca = CA_default # The default ca section
[ CA_default ]
dir = /localca # Where everything is kept
certs = $dir/certs # Where the issued certs are kept
crl_dir = $dir/crl # Where the issued crl are kept
database = $dir/index.txt # database index file.
new_certs_dir = $dir/ # default place for new certs.
certificate = $dir/cacert.pem # The CA certificate
serial = $dir/serial # The current serial number
crlnumber = $dir/crlnumber
crl = $dir/crl.pem # The current CRL
private_key = $dir/private/cakey.pem # The private key
x509_extensions = usr_cert # The extensions to add to the cert
name_opt = ca_default # Subject Name options
cert_opt = ca_default # Certificate field options
default_days = 365 # how long to certify for
default_crl_days= 30 # how long before next CRL
default_md = default # use public key default MD
preserve = no # keep passed DN ordering
policy = policy_match
[ policy_match ]
countryName = optional
stateOrProvinceName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
[ policy_anything ]
countryName = optional
stateOrProvinceName = optional
localityName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
```

openssl.cnf

```
emailAddress = optional
[ req ]
default_bits = 2048
default_keyfile = privkey.pem
distinguished_name = req_distinguished_name
attributes = req_attributes
x509_extensions = v3_ca # The extensions to add to the self signed cert
string_mask = utf8only
[ req_distinguished_name ]
countryName = Country Name (2 letter code)
countryName_default = AU
countryName_min = 2
countryName_max = 2
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Some-State
localityName = Locality Name (eg, city)
0.organizationName = Organization Name (eg, company)
0.organizationName_default = Internet Widgits Pty Ltd
organizationalUnitName = Organizational Unit Name (eg, section)
commonName = Common Name (e.g. server FQDN or YOUR name)
commonName_max = 64
emailAddress = Email Address
emailAddress_max = 64
[ req_attributes ]
challengePassword = A challenge password
challengePassword_min = 4
challengePassword_max = 20
unstructuredName = An optional company name
[ usr_cert ]
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
[ v3_ca ]
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid:always,issuer
basicConstraints = critical,CA:true
[ v3_ca_reqexts ]
subjectKeyIdentifier = hash
keyUsage = digitalSignature, keyCertSign, cRLSign
[ v3_issuing_extensions ]
subjectKeyIdentifier = hash
keyUsage = digitalSignature, keyCertSign, cRLSign
basicConstraints = critical,CA:TRUE
authorityKeyIdentifier = keyid:always,issuer:always
[ v3_ca_reqexts ]
subjectKeyIdentifier = hash
keyUsage = digitalSignature, keyCertSign, cRLSign
```

openssl.cnf

```
[ v3_datavault_reqexts ]
subjectKeyIdentifier = hash
keyUsage = keyEncipherment
extendedKeyUsage = emailProtection
[ v3_scep_reqexts ]
subjectKeyIdentifier = hash
[ v3_web_reqexts ]
subjectKeyIdentifier = hash
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
[ v3_ca_extensions ]
subjectKeyIdentifier = hash
keyUsage = digitalSignature, keyCertSign, cRLSign
basicConstraints = critical,CA:TRUE
authorityKeyIdentifier = keyid:always,issuer
[ v3_issuing_extensions ]
subjectKeyIdentifier = hash
keyUsage = digitalSignature, keyCertSign, cRLSign
basicConstraints = critical,CA:TRUE
authorityKeyIdentifier = keyid:always,issuer:always
[ v3_datavault_extensions ]
subjectKeyIdentifier = hash
keyUsage = keyEncipherment
extendedKeyUsage = emailProtection
basicConstraints = CA:FALSE
authorityKeyIdentifier = keyid:always,issuer
[ v3_scep_extensions ]
subjectKeyIdentifier = hash
keyUsage = digitalSignature, keyEncipherment
basicConstraints = CA:FALSE
authorityKeyIdentifier = keyid,issuer
[ v3_web_extensions ]
subjectKeyIdentifier = hash
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
basicConstraints = critical,CA:FALSE
subjectAltName = DNS:Openxpki
[ crl_ext ]
authorityKeyIdentifier=keyid:always
[ proxy_cert_ext ]
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
proxyCertInfo=critical,language:id-ppl-anyLanguage,pathlen:3,policy:foo
[ tsa ]
default_tsa = tsa_config1 # the default TSA section
[ tsa_config1 ]
dir = ./demoCA # TSA root directory
serial = $dir/tsaserial # The current serial number
(mandatory)
```

openssl.cnf

```
crypto_device = builtin # OpenSSL engine to use for signing
signer_cert = $dir/tsacert.pem # The TSA signing certificate
certs = $dir/cacert.pem
signer_key = $dir/private/tsakey.pem # The TSA private key (optional)
signer_digest = sha256 # Signing digest to use. (Optional)
default_policy = tsa_policy1 # Policy if request did not specify it
 # (optional)
 other_policies = tsa_policy2, tsa_policy3 # acceptable policies
(optional)
digests = sha1, sha256, sha384, sha512 # Acceptable message digests
(mandatory)
accuracy = secs:1, millisecs:500, microsecs:100 # (optional)
clock_precision_digits = 0 # number of digits after dot. (optional)
ordering = yes # Is ordering defined for timestamps?
 # (optional, default: no)
tsa_name = yes # Must the TSA name be included in the reply?
 # (optional, default: no)
ess_cert_id_chain = no # Must the ESS cert id chain be included?
 # (optional, default: no)
ess_cert_id_alg = sha1 # algorithm to compute certificate
 # identifier (optional, default: sha1)
[insta] # CMP using Insta Demo CA
# Message transfer
server = pki.certificate.fi:8700
path = pkix/
recipient = "/C=FI/O=Insta Demo/CN=Insta Demo CA" # or set srvcert or issuer
ignore_keyusage = 1 # potentially needed quirk
unprotected_errors = 1 # potentially needed quirk
extracertsout = insta.extracerts.pem
ref = 3078 # user identification
secret = pass:insta # can be used for both client and server side
cmd = ir # default operation, can be overridden on cmd line with, e.g., kur
subject = "/CN=openssl-cmp-test"
newkey = insta.priv.pem
out_trusted = apps/insta.ca.crt # does not include keyUsage digitalSignature
certout = insta.cert.pem
[pbm]
ref = $insta::ref # 3078
secret = $insta::secret # pass:insta
[signature] # Signature-based protection for Insta CA
trusted = $insta::out_trusted # apps/insta.ca.crt
secret = # disable PBM
key = $insta::newkey # insta.priv.pem
cert = $insta::certout # insta.cert.pem
[ir]
cmd = ir
[cr]
cmd = cr
[kur]
cmd = kur
```

openssl.cnf

```
oldcert = $insta::certout # insta.cert.pem
[rr]
cmd = rr
oldcert = $insta::certout # insta.cert.pem
[openssl_init]
engines=engine_section
[engine_section]
pkcs11 = pkcs11_section
[pkcs11_section]
engine_id = pkcs11
dynamic_path = /usr/lib/x86_64-linux-gnu/engines-3/pkcs11.so
MODULE_PATH = /opt/utimaco/lib/libcs_pkcs11_R3.so
init = 0
```

5.2.2 Verify PKCS#11 Engine

Run the command below to verify if the OpenSSL Engine is available or not.

›_ Console

```
# openssl engine pkcs11 -t
```

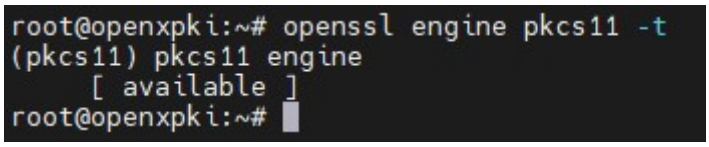

Figure 6 : Verify if engine is available

5.3 Installing OpenXPKI and Dependent Packages

1. Update the apt repository.

›_ Console

```
# apt-get update
```

```
root@openxpki:~# apt-get update
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [52.1 kB]
Get:3 http://security.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main Sources [9,640 kB]
Get:5 http://security.debian.org/debian-security bookworm-security/main Sources [45.2 kB]
Get:6 http://security.debian.org/debian-security bookworm-security/non-free-firmware Sources [784 B]
Get:7 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [60.8 kB]
Get:8 http://security.debian.org/debian-security bookworm-security/main Translation-en [36.4 kB]
Get:9 http://security.debian.org/debian-security bookworm-security/non-free-firmware amd64 Packages [680 B]
Get:10 http://security.debian.org/debian-security bookworm-security/non-free-firmware Translation-en [464 B]
Get:11 http://deb.debian.org/debian bookworm/non-free-firmware Sources [6,156 B]
Get:12 http://deb.debian.org/debian bookworm/main amd64 Packages [8,906 kB]
Get:13 http://deb.debian.org/debian bookworm/main Translation-en [6,078 kB]
Get:14 http://deb.debian.org/debian bookworm/non-free-firmware amd64 Packages [6,240 B]
Get:15 http://deb.debian.org/debian bookworm-updates/main Sources [1,748 B]
Get:16 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [4,952 B]
Get:17 http://deb.debian.org/debian bookworm-updates/main Translation-en [3,632 B]
Fetched 25.0 MB in 4s (6,293 kB/s)
Reading package lists... Done
N: Repository 'http://deb.debian.org/debian bookworm InRelease' changed its 'Version' value from '12.0' to '12.1'
root@openxpki:~#
```

Figure 7 : Update apt repository

2. Add openxpki package signing key.

›_ Console

```
# wget https://packages.openxpki.org/v3/bookworm/Release.key -O - 2>/dev/null

| tee Release.key | gpg -o /usr/share/keyrings/openxpki.pgp –dearmor
```

```
root@openxpki:~# wget https://packages.openxpki.org/v3/bookworm/Release.key -O - 2>/dev/null | tee Release.key | gpg -o /usr/s
hare/keyrings/openxpki.pgp --dearmor
root@openxpki:~#
```

Figure 8 : Add signing key

3. Add the openxpki repository to your source list and update.

›_ Console

```
# echo -e "Types: deb\nURIs: https://packages.openxpki.org/v3/bookworm/\nSuites:
bookworm\nComponents: release\nSigned-By: /usr/share/keyrings/openxpki.pgp" > /
etc/apt/sources.list.d/openxpki.sources # apt update
```

```
root@openxpki:~# echo -e "Types: deb\nURIs: https://packages.openxpki.org/v3/bookworm/\nSuites: bookworm\nComponents: release\nSigned-By:
/usr/share/keyrings/openxpki.pgp" > /etc/apt/sources.list.d/openxpki.sources
apt update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://security.debian.org/debian-security bookworm-security InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Get:4 https://packages.openxpki.org/v3/bookworm bookworm InRelease [1,881 B]
Get:5 https://packages.openxpki.org/v3/bookworm bookworm/release amd64 Packages [5,169 B]
Fetched 7,050 B in 1s (5,115 B/s)
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
34 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@openxpki:~#
```

Figure 9 : Add openxpki repository

4. Install mariadb-server and libdbd-mariadb-perl.

›_ Console

```
# apt install mariadb-server libdbd-mariadb-perl
```

```
root@openxpki:~# apt install mariadb-server libdbd-mariadb-perl
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  galera-4 gawk libcgi-fast-perl libcgi-pm-perl libclone-perl libconfig-inifiles-perl libdaxctl1 libdbi-perl libencode-locale-perl
  libfcgi-bin libfcgi-perl libfcgi0ldbl libgpm2 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl liblzo2-2 libmariadb3 libmpfr6 libncurses6 libndctl6 libnuma1 libpmem1
  libregexp-ipv6-perl libsigsegv2 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl liburing2 mariadb-client
  mariadb-client-core mariadb-common mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4 mariadb-plugin-provider-lzma
  mariadb-plugin-provider-lzo mariadb-plugin-provider-snappy mariadb-server-core mysql-common psmisc pv rsync socat
Suggested packages:
  gawk-doc libmldbm-perl libnet-daemon-perl libsql-statement-perl gpm libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl
  libwww-perl mailx mariadb-test netcat-openbsd doc-base python3-braceexpand
The following NEW packages will be installed:
  galera-4 gawk libcgi-fast-perl libcgi-pm-perl libclone-perl libconfig-inifiles-perl libdaxctl1 libdbd-mariadb-perl libdbi-perl
  libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldbl libgpm2 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl liblzo2-2 libmariadb3 libmpfr6 libncurses6 libndctl6
  libnuma1 libpmem1 libregexp-ipv6-perl libsigsegv2 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl liburing2
  mariadb-client mariadb-client-core mariadb-common mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4
  mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo mariadb-plugin-provider-snappy mariadb-server mariadb-server-core
  mysql-common psmisc pv rsync socat
0 upgraded, 50 newly installed, 0 to remove and 34 not upgraded.
Need to get 20.5 MB of archives.
After this operation, 197 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bookworm/main amd64 libmpfr6 amd64 4.2.0-1 [701 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 libsigsegv2 amd64 2.14-1 [37.2 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 gawk amd64 1:5.2.1-2 [673 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 mysql-common all 5.8+1.1.0 [6,636 B]
Get:5 http://deb.debian.org/debian bookworm/main amd64 mariadb-common all 1:10.11.3-1 [24.0 kB]
Get:6 http://deb.debian.org/debian bookworm/main amd64 galera-4 amd64 26.4.13-1 [825 kB]
Get:7 http://deb.debian.org/debian bookworm/main amd64 libdbi-perl amd64 1.643-4 [773 kB]
Get:8 http://deb.debian.org/debian bookworm/main amd64 libconfig-inifiles-perl all 3.000003-2 [45.9 kB]
Get:9 http://deb.debian.org/debian bookworm/main amd64 libmariadb3 amd64 1:10.11.3-1 [172 kB]
```

Figure 10 : MariaDB installation

```
Setting up libencode-locale-perl (1.05-3) ...
Setting up libsnappy1v5:amd64 (1.1.9-3) ...
Setting up socat (1.7.4.4-2) ...
Setting up libncurses6:amd64 (6.4-4) ...
Setting up libio-html-perl (1.004-3) ...
Setting up libmariadb3:amd64 (1:10.11.3-1) ...
Setting up libdaxctl1:amd64 (76.1-1) ...
Setting up libtimedate-perl (2.3300-2) ...
Setting up libregexp-ipv6-perl (0.03-3) ...
Setting up libnuma1:amd64 (2.0.16-1) ...
Setting up pv (1.6.20-1) ...
Setting up libndctl6:amd64 (76.1-1) ...
Setting up libfcgi-perl (0.82+ds-2) ...
Setting up libterm-readkey-perl (2.38-2+b1) ...
Setting up liburing2:amd64 (2.3-3) ...
Setting up libpmem1:amd64 (1.12.1-2) ...
Setting up liburi-perl (5.17-1) ...
Setting up libdbi-perl:amd64 (1.643-4) ...
Setting up rsync (3.2.7-1) ...
rsync.service is a disabled or a static unit, not starting it.
Setting up libhttp-date-perl (6.05-2) ...
Setting up mariadb-client-core (1:10.11.3-1) ...
Setting up libdbd-mariadb-perl (1.22-1+b1) ...
Setting up libhtml-parser-perl:amd64 (3.81-1) ...
Setting up mariadb-server-core (1:10.11.3-1) ...
Setting up libhttp-message-perl (6.44-1) ...
Setting up mariadb-client (1:10.11.3-1) ...
Setting up libcgi-pm-perl (4.55-1) ...
Setting up libhtml-template-perl (2.97-2) ...
Setting up mariadb-server (1:10.11.3-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /lib/systemd/system/mariadb.service.
Setting up mariadb-plugin-provider-bzip2 (1:10.11.3-1) ...
Setting up mariadb-plugin-provider-lzma (1:10.11.3-1) ...
Setting up mariadb-plugin-provider-lzo (1:10.11.3-1) ...
Setting up mariadb-plugin-provider-lz4 (1:10.11.3-1) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Setting up mariadb-plugin-provider-snappy (1:10.11.3-1) ...
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9) ...
Processing triggers for mariadb-server (1:10.11.3-1) ...
root@openxpki:~#
```

Figure 11 : MariaDB installation

5. Install apache2 and fastcgi module.

›_ Console

```
# apt install apache2 libapache2-mod-fcgid
```

```
root@openxpki:~# apt install apache2 libapache2-mod-fcgid
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl3-gnutls libcurl4 liblua5.3-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-fcgid libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libcurl4
  liblua5.3-0 ssl-cert
The following packages will be upgraded:
  libcurl3-gnutls
1 upgraded, 12 newly installed, 0 to remove and 33 not upgraded.
Need to get 3,138 kB of archives.
After this operation, 9,382 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://security.debian.org/debian-security bookworm-security/main amd64 libcurl4 amd64 7.88.1-10+deb12u1 [386 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 libapr1 amd64 1.7.2-3 [102 kB]
Get:3 http://security.debian.org/debian-security bookworm-security/main amd64 libcurl3-gnutls amd64 7.88.1-10+deb12u1 [381 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 libaprutil1 amd64 1.6.3-1 [87.8 kB]
Get:5 http://deb.debian.org/debian bookworm/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1 [13.6 kB]
Get:6 http://deb.debian.org/debian bookworm/main amd64 libaprutil1-ldap amd64 1.6.3-1 [11.8 kB]
Get:7 http://deb.debian.org/debian bookworm/main amd64 liblua5.3-0 amd64 5.3.6-2 [123 kB]
Get:8 http://deb.debian.org/debian bookworm/main amd64 apache2-bin amd64 2.4.57-2 [1,364 kB]
Get:9 http://deb.debian.org/debian bookworm/main amd64 apache2-data all 2.4.57-2 [160 kB]
Get:10 http://deb.debian.org/debian bookworm/main amd64 apache2-utils amd64 2.4.57-2 [202 kB]
Get:11 http://deb.debian.org/debian bookworm/main amd64 apache2 amd64 2.4.57-2 [215 kB]
Get:12 http://deb.debian.org/debian bookworm/main amd64 libapache2-mod-fcgid amd64 1:2.3.9-4 [71.2 kB]
Get:13 http://deb.debian.org/debian bookworm/main amd64 ssl-cert all 1.1.2 [21.1 kB]
Fetched 3,138 kB in 1s (4,859 kB/s)
Reading changelogs ... Done
Preconfiguring packages ...
Selecting previously unselected package libapr1:amd64.
(Reading database ... 31924 files and directories currently installed.)
Preparing to unpack .../00-libapr1_1.7.2-3_amd64.deb ...
Unpacking libapr1:amd64 (1.7.2-3) ...
Selecting previously unselected package libaprutil1:amd64.
Preparing to unpack .../01-libaprutil1_1.6.3-1_amd64.deb ...
```

Figure 12 : Apache2 installation

```
Setting up libaprutil1-ldap:amd64 (1.6.3-1) ...
Setting up libaprutil1-dbd-sqlite3:amd64 (1.6.3-1) ...
Setting up apache2-utils (2.4.57-2) ...
Setting up apache2-bin (2.4.57-2) ...
Setting up libapache2-mod-fcgid (1:2.3.9-4) ...
Package apache2 is not configured yet. Will defer actions by package libapache2-mod-fcgid.
Setting up apache2 (2.4.57-2) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
info: Executing deferred 'a2enmod fcgid' for package libapache2-mod-fcgid
Enabling module fcgid.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9) ...
root@openxpki:~#
```

Figure 13 : Apache2 installation

6. Enable fcgid module.

›_ Console

```
# a2enmod fcgid
```

```
root@openxpki:~# a2enmod fcgid
Enabling module fcgid.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@openxpki:~#
```

Figure 14 : Enable fcgid

Fastcgi module should be enabled explicitly, otherwise, the .fcgi file will be treated as plain text (this is usually done by the installer already):

7. Install the OpenXPKI core package, session driver and translation package.

›_ Console

```
# apt install libopenxpki-perl openxpki-cgi-session-driver openxpki-i18n
```

```
root@openxpki:~# apt install libopenxpki-perl openxpki-cgi-session-driver openxpki-i18n
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  cpp cpp-12 libalgorithm-c3-perl libappconfig-perl libarchive-zip-perl libauthen-sasl-perl libb-hooks-endofscope-perl
  libb-hooks-op-check-perl libcache-lru-perl libcarp-clan-perl libcgi-session-perl libclass-accessor-perl libclass-c3-perl
  libclass-c3-xs-perl libclass-data-inheritable-perl libclass-factory-perl libclass-inspector-perl libclass-isa-perl libclass-load-perl
  libclass-load-xs-perl libclass-method-modifiers-perl libclass-observable-perl libclass-singleton-perl libclass-std-perl
  libclass-tiny-perl libclass-xsaccessor-perl libclone-choose-perl libcommon-sense-perl libconfig-any-perl libconfig-general-perl
  libconfig-merge-perl libconfig-std-perl libconfig-tiny-perl libconnector-perl libconvert-asn1-perl libconvert-binhex-perl
  libcrypt-argon2-perl libcrypt-cbc-perl libcrypt-jwt-perl libcrypt-openssl-aes-perl libcrypt-pbkdf2-perl libcrypt-pkcs10-perl
  libcrypt-rijndael-perl libcrypt-smime-perl libcrypt-x509-perl libcryptx-perl libdata-dump-perl libdata-optlist-perl
  libdata-serializer-perl libdata-stream-bulk-perl libdatetime-format-dateparse-perl libdatetime-format-strptime-perl
  libdatetime-locale-perl libdatetime-perl libdatetime-timezone-perl libdbd-pg-perl libdbix-handler-perl libdbix-transactionmanager-perl
  libdevel-callchecker-perl libdevel-caller-perl libdevel-globaldestruction-perl libdevel-lexalias-perl libdevel-overloadinfo-perl
  libdevel-partialdump-perl libdevel-stacktrace-perl libdevel-symdump-perl libdigest-bubblebabble-perl libdigest-hmac-perl
  libdigest-sha3-perl libdist-checkconflicts-perl libdynaloader-functions-perl libemail-date-format-perl libencode-perl
  libeval-closure-perl libexception-class-perl libexporter-tiny-perl libfeature-compat-try-perl libfile-find-rule-perl
  libfile-listing-perl libfile-sharedir-perl libfile-slurp-perl libfile-slurp-tiny-perl libfilter-perl libfont-afm-perl
  libfreezethaw-perl libgetopt-long-descriptive-perl libgssapi-perl libhash-merge-perl libhtml-form-perl libhtml-format-perl
  libhtml-tree-perl libhttp-cookies-perl libhttp-daemon-perl libhttp-negotiate-perl libimport-into-perl libintl-perl libintl-xs-perl
  libio-digest-perl libio-multiplex-perl libio-prompt-perl libio-sessiondata-perl libio-socket-inet6-perl libio-socket-ssl-perl
  libio-stringy-perl libipc-shareable-perl libisl23 libjson-perl libjson-xs-perl liblist-moreutils-perl liblist-moreutils-xs-perl
  liblog-dispatch-perl liblog-log4perl-perl liblwp-protocol-https-perl libmail-rfc822-address-perl libmail-sendmail-perl
  libmailtools-perl libmath-int128-perl libmath-int64-perl libmime-lite-perl libmime-tools-perl libmime-types-perl
  libmodule-implementation-perl libmodule-pluggable-perl libmodule-runtime-conflicts-perl libmodule-runtime-perl libmoo-perl
  libmoose-perl libmoosex-getopt-perl libmoosex-insideout-perl libmoosex-nonmoose-perl libmoosex-params-validate-perl
  libmoosex-role-parameterized-perl libmoosex-strictconstructor-perl libmoosex-types-path-class-perl libmoosex-types-perl libmpc3
  libmro-compat-perl libnamespace-autoclean-perl libnamespace-clean-perl libnet-cidr-perl libnet-dns-perl libnet-dns-sec-perl
  libnet-http-perl libnet-ip-perl libnet-ldap-perl libnet-libidn2-perl libnet-server-perl libnet-smtp-ssl-perl libnet-ssleay-perl
  libnetaddr-ip-perl libnumber-compare-perl libossp-uuid-perl libossp-uuid16 libpackage-deprecationmanager-perl libpackage-stash-perl
  libpackage-stash-xs-perl libpadwalker-perl libparams-classify-perl libparams-util-perl libparams-validate-perl
  libparams-validationcompiler-perl libpath-class-perl libperl4-corelibs-perl libperlio-via-dynamic-perl libpod-coverage-perl
  libpod-parser-perl libpod-pom-perl libpq5 libproc-processtable-perl libproc-safeexec-perl libreadonly-perl libref-util-perl
  libref-util-xs-perl libregexp-common-perl librole-tiny-perl libsoap-lite-perl libsocket6-perl libspecio-perl
```

Figure 15 : OpenXPKI installation

```
Setting up libmoosex-strictconstructor-perl (0.21-2) ...
Setting up libcrypt-pbkdf2-perl (0.161520-2) ...
Setting up libdevel-partialdump-perl (0.20-2) ...
Setting up libdatetime-timezone-perl (1:2.60-1+2023c) ...
Setting up libmoosex-types-perl (0.50-2) ...
Setting up libcrypt-cbc-perl (3.04-3) ...
Setting up liblog-dispatch-perl (2.70-1) ...
Setting up libmoosex-types-path-class-perl (0.09-2) ...
Setting up libdatetime-perl (2:1.59-1) ...
Setting up libdatetime-format-strptime-perl (1.7900-1) ...
Setting up libdatetime-format-dateparse-perl (0.05-3) ...
Setting up libworkflow-perl (1.62-1) ...
Setting up libwww-perl (6.68-1) ...
Setting up liblwp-protocol-https-perl (6.10-1) ...
Setting up libxml-parser-perl (2.46-4) ...
Setting up libxml-sax-expat-perl (0.51-2) ...
update-perl-sax-parsers: Registering Perl SAX parser XML::SAX::Expat with priority 50...
update-perl-sax-parsers: Updating overall Perl SAX parser modules info file...
Replacing config file /etc/perl/XML/SAX/ParserDetails.ini with new version
Setting up libsoap-lite-perl (1.27-3) ...
Setting up libxmlrpc-lite-perl (0.717-5) ...
Setting up libopenxpki-perl (3.26.0-0) ...
OpenXPKI Password for Test Accounts was set to ##apjTYiU6QuOo##
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
Enabling site openxpki.
To activate the new configuration, you need to run:
  systemctl reload apache2
Processing triggers for libc-bin (2.36-9) ...
Processing triggers for man-db (2.11.2-2) ...
root@openxpki:~#
```

Figure 16 : OpenXPKI installation

8. Use the openxpkiadm command to verify if the openxpki was installed correctly.

›_ Console

```
# openxpkiadm version
```

Figure 17 : OpenXPKI version

9. Create an empty database and assign a database user.

›_ Console

```
# mariadb

> CREATE DATABASE openxpki CHARSET utf8;

> CREATE USER 'openxpki'@'localhost' IDENTIFIED BY 'openxpki';

> GRANT ALL ON openxpki.* TO 'openxpki'@'localhost'; > flush privileges;
```

```
root@openxpki:~# mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.3-MariaDB-1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE openxpki CHARSET utf8;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'openxpki'@'localhost' IDENTIFIED BY 'openxpki';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> GRANT ALL ON openxpki.* TO 'openxpki'@'localhost';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

Figure 18 : Database creation

10. Enter the above database credentials into `/etc/openxpki/config.d/system/database.yaml` file as shown below.

database.yaml

```
main:
 debug: 0
 type: MariaDB
 name: openxpki
 user: openxpki
 passwd: openxpki
```

```
root@openxpki:~# cat /etc/openxpki/config.d/system/database.yaml
main:
    debug: 0
    # This requires libdbd-mysql-perl - see notes regarding driver at
    # https://openxpki.readthedocs.io/en/latest/reference/configuration/server.html
    type: MariaDB
    name: openxpki
    #host: localhost
    #port: 3306
    user: openxpki
    passwd: openxpki
```

Figure 19 : Database credentials

For OpenXPKI version 3.26.1 you need to replace type value MariaDB to MariabDB2 in database.yaml file.

11. Create the empty database schema from the provided schema file `/usr/share/doc/libopenxpki-perl/examples/schema-mariadb.sql`.

›_ Console

```
# cat /usr/share/doc/libopenxpki-perl/examples/schema-mariadb.sql | mysql -u
root --password --database  openxpki
```

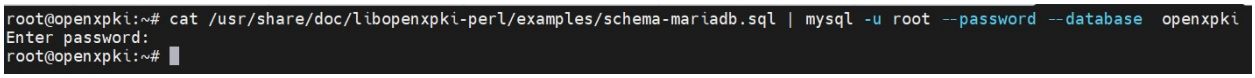

Figure 20 : Create database schema

Enter the database user password when prompted.

12. Update `/etc/apache2/ports.conf` with any http port.

ports.conf

```
…
Listen 8080
…
```

13. Update `/etc/apache2/sites-enabled/openxpki.conf` with below entries.

openxpki.conf

```
<VirtualHost *:8080>
 ServerAlias *
 DocumentRoot /var/www/
 ScriptAlias /rpc /usr/lib/cgi-bin/rpc.fcgi
 ScriptAlias /healthcheck /usr/lib/cgi-bin/healthcheck.fcgi
 ScriptAlias /certep /usr/lib/cgi-bin/certep.fcgi
 ScriptAlias /.well-known/est /usr/lib/cgi-bin/est.fcgi
 ScriptAlias /cmc /usr/lib/cgi-bin/cmc.fcgi
 ScriptAliasMatch ^/(webui/)?(([a-z0-9-]+)/)?cgi-bin/webui.fcgi
/usr/lib/cgi-bin/webui.fcgi
 RewriteEngine On
 RewriteRule ^/$ https://%{HTTP_HOST}/webui/index/ [L,R=301,NC]
 RewriteRule ^/([a-z0-9-]+)$ https://%{HTTP_HOST}/$1/ [L,R=301,NC]
 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_FILENAME} !-d
 RewriteRule ^/([a-z0-9-]+)/$ /var/www/openxpki/index.html [L]
 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_FILENAME} !-f
 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_FILENAME} !-d
 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_FILENAME} !-l
 RewriteCond %{REQUEST_FILENAME} !(cgi
bin|rpc|cmc|certep|download|healthcheck)
 RewriteRule ^/(webui/)?([a-z0-9-]+)/(.*) /var/www/openxpki/$3 [L,NC]
 <IfModule mod_headers.c>
 Header set Strict-Transport-Security max-age=31536000
 Header set X-Frame-Options deny
 Header set X-XSS-Protection "1; mode=block;"
 </IfModule>
</VirtualHost>
```

It is highly recommended to run OpenXPKI Web UI on HTTPS. If you want OpenXPKI Web UI to run over HTTPS, then you can modify the above file with SSL settings. You can generate SSL keys on Utimaco HSM and integrate it by downloading and following the Apache Integration guide with Utimaco HSM from the support portal.

14. Reload and restart apche2 service.

›_ Console

```
# systemctl reload apache2.service

# systemctl restart apache2.service
```

```
root@openxpki:~# systemctl reload apache2.service
root@openxpki:~#
root@openxpki:~# systemctl restart apache2.service
root@openxpki:~#
```

Figure 21 : Reload and restart apache2 service

5.4 Generating a Root CA Key and Certificate

1. Open the `/< OPENSSLDIR>/openssl.cnf` file in the text editor and edit the `[CA_default]` section to following.

openssl.cnf

```
. . .
dir = /localCA
new_certs_dir = $dir/certs
. . .
```

You can change dir to the directory of your choice but make sure to use correct path in the subsequent steps. Here we have created the directory `/localCA` under root directory and `new_certs_dir= $dir/certs`

2. Create the directory `/localCA/certs`.

›_ Console

```
# mkdir /localCA/certs
```

3. Create the text files `/localCA/index.txt` and `/localCA/serial`.

›_ Console

```
# touch /localCA/index.txt

# touch /localCA/serial
```

4. Open the `/localCA/serial` file and write 01 in it and click enter. Save the file.
5. Create a key pair by using p11tool2 for root CA.

›_ Console

```
# p11tool2 slot=<Slot_No.> LoginUser=ask PubKeyAttr=CKA_LABEL="CAKey"

PrvKeyAttr=CKA_LABEL="CAKey" GenerateKeyPair=RSA
```

This generates RSA 2048 CA private and public keys on the HSM.

6. Verify that the RSA keys are generated onto the HSM using the following command.

›_ Console

```
# p11tool2 Slot=<Slot_No.> loginUser=ask listobjects
```

```
root@openxpki:~# p11tool2 slot=33 loginuser=ask listobjects
Enter normal user PIN:


CKO_PUBLIC_KEY:

+ 1.1
  CKA_KEY_TYPE                    = CKK_RSA
  CKA_UNIQUE_ID                   = 97AF319C-E68E-47EE-B73C-433E7285D957
  CKA_LABEL                       = CAKey
  CKA_ID                          =



CKO_PRIVATE_KEY:

+ 2.1
  CKA_KEY_TYPE                    = CKK_RSA
  CKA_UNIQUE_ID                   = 05341290-AC7D-4980-9A22-9E64AC5E066E
  CKA_SENSITIVE                   = CK_TRUE
  CKA_EXTRACTABLE                 = CK_FALSE
  CKA_LABEL                       = CAKey
  CKA_ID                          =

root@openxpki:~#
```

Figure 22 : List keys

7. Create the CA certificate based on the generated key.

›_ Console

```
# openssl req -x509 -engine pkcs11 -keyform engine  -extensions v3_ca_extensions
-batch  -new -key "pkcs11:token=openxpki;object=CAKey" sha256 -out RootCA.pem
-subj "/CN=OpenXPKI Root CA 20230926"
```

```
root@openxpki:/localCA# openssl req -x509 -engine pkcs11 -keyform engine  -extensions v3_ca_extensions -batch  -new -key "pkcs
11:token=openxpki;object=CAKey" -sha256 -out RootCA.pem  -subj "/CN=OpenXPKI Root CA 20230926"
Engine "pkcs11" set.
Enter PKCS#11 token PIN for openxpki:
root@openxpki:/localCA#
```

Figure 23 : RootCA certificate generation

Here **CAKey** is the Object label for the CA private key on the Utimaco HSM created in Step 5 and **openxpki** is a token label. Provide Slot PIN when prompted.

5.5 Generating a Subordinate CA Key and Certificate

1. Generate a Subordinate CA key pair using p11tool2.

›_ Console

```
# p11tool2 slot=<Slot_No.> LoginUser=<ask> PubKeyAttr=CKA_LABEL=Subca

PrvKeyAttr=CKA_LABEL=Subca GenerateKeyPair=RSA
```

2. Verify that the keys are generated onto the HSM using the following command.

›_ Console

```
# p11tool2 slot=<Slot_No.> loginUser=<ask> listobjects
```

```
root@openxpki:~# p11tool2 slot=33 loginuser=ask listobjects
Enter normal user PIN:


CKO_PUBLIC_KEY:

+ 1.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 720C04E9-4E38-4996-BEBB-615C948D7138
  CKA_LABEL                     = Subca
  CKA_ID                        =


+ 1.2
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 97AF319C-E68E-47EE-B73C-433E7285D957
  CKA_LABEL                     = CAKey
  CKA_ID                        =



CKO_PRIVATE_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = A94046A5-B75D-4E70-A276-DC98E2E04F76
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = Subca
  CKA_ID                        =


+ 2.2
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 05341290-AC7D-4980-9A22-9E64AC5E066E
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = CAKey
  CKA_ID                        =

root@openxpki:~#
```

Figure 24 : List keys

3. Generate a certificate request for Subordinate CA.

›_ Console

```
# openssl req -engine pkcs11 -reqexts v3_ca_reqexts -batch -new -key

"pkcs11:token=openxpki;object=Subca" -keyform engine -out SubCA.csr -subj

"/C=DE/O=OpenXPKI/OU=PKI/CN=OpenXPKI Demo Issuing CA 20230926"
```

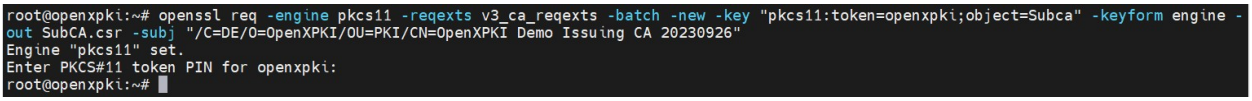

Figure 25 : Subordinate CA CSR generation

Here **openxpki** is the token label and **Subca** is the key on the HSM. Provide Slot PIN when prompted.

4. Sign the certificate request for Subordinate CA by Root CA.

›_ Console

```
# openssl ca -create_serial -engine pkcs11 -extensions v3_issuing_extensions
batch -cert RootCA.pem -in SubCA.csr -keyfile

"pkcs11:token=openxpki;object=CAKey" -keyform engine -out SubCA.pem
```

```
root@openxpki:/localCA# openssl ca -create_serial -engine pkcs11 -extensions v3_issuing_extensions -batch -cert RootCA.pem -in
 SubCA.csr -keyfile "pkcs11:token=openxpki;object=CAKey" -keyform engine -out SubCA.pem
Engine "pkcs11" set.
Using configuration from /usr/lib/ssl/openssl.cnf
Enter PKCS#11 token PIN for openxpki:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Sep 26 08:30:30 2023 GMT
            Not After : Sep 25 08:30:30 2024 GMT
        Subject:
            countryName               = DE
            organizationName          = OpenXPKI
            organizationalUnitName    = PKI
            commonName                = OpenXPKI Demo Issuing CA 20230926
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                20:40:29:BE:98:69:4B:D1:20:4A:65:30:A1:49:54:59:9C:A6:4B:27
            X509v3 Key Usage:
                Digital Signature, Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Authority Key Identifier:
                keyid:5D:1B:9B:87:7A:ED:C6:8D:C5:5C:CA:7A:FB:D6:BD:EE:60:BE:4D:CB
                DirName:/CN=OpenXPKI Root CA 20230926
                serial:5C:90:0A:79:DE:34:F0:1A:0F:9C:7A:29:7B:21:AB:8F:5E:78:AC:B0
Certificate is to be certified until Sep 25 08:30:30 2024 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
root@openxpki:/localCA#
```

Figure 26 : SubCA CSR signing

5.6 Generating DataVault Key and Certificate

1. Generate a DataVault key pair using p11tool2.

›_ Console

```
# p11tool2 slot=33 LoginUser=ask PubKeyAttr=CKA_LABEL=Datavault

PrvKeyAttr=CKA_LABEL=Datavault GenerateKeyPair=RSA
```

```
root@openxpki:/localCA# p11tool2 slot=33 LoginUser=ask PubKeyAttr=CKA_LABEL=Datavault PrvKeyAttr=CKA_LABEL=Datavault GenerateK
eyPair=RSA
Enter normal user PIN:
```

Figure 27 : Generating DataVault key

2. Verify that the keys are generated onto the HSM using the following command.

›_ Console

```
# p11tool2 slot=<Slot_No.> loginUser=<ask> listobjects
```

```
root@openxpki:/localCA# p11tool2 slot=33 loginuser=ask listobjects
Enter normal user PIN:


CKO_PUBLIC_KEY:

+ 1.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 042D09DA-824C-49F3-AF76-27A16373C537
  CKA_LABEL                     = Subca
  CKA_ID                        =


+ 1.2
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = D15521BA-75E8-477F-A23E-2E61BF491CFF
  CKA_LABEL                     = CAKey
  CKA_ID                        =


+ 1.3
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 108132FF-297F-45A9-AD07-543B78CC6600
  CKA_LABEL                     = Datavault
  CKA_ID                        =



CKO_PRIVATE_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 881868CE-53EC-421E-90F8-ADEC67607C8A
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = Subca
  CKA_ID                        =


+ 2.2
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 412FFD73-6843-4E71-980A-744A39D05698
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = Datavault
  CKA_ID                        =


+ 2.3
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 6D3094F7-F6E3-4B40-9924-EC9359B235BE
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = CAKey
```

Figure 28 : List keys

3. Generate a self-signed certificate for DataVault.

›_ Console

```
# openssl req -x509 -engine pkcs11 -keyform engine -extensions

v3_datavault_extensions -batch -new -key "pkcs11:token=openxpki;obj
ect=Datavault" -sha256 -subj "/CN=DataVault" -out "OpenXPKI_DataVault.crt"
```

```
root@openxpki:/localCA# openssl req -x509 -engine pkcs11 -keyform engine -extensions v3_datavault_extensions -batch -new -key "pkcs11:token=openxpki;obj
ect=Datavault" -sha256 -subj "/CN=DataVault" -out "OpenXPKI_DataVault.crt"
Engine "pkcs11" set.
Enter PKCS#11 token PIN for openxpki:
root@openxpki:/localCA#
```

Figure 29 : Generating a self-signed DataVault certificate

Here **openxpki** is the token label and **Datavault** is the key on the HSM. Provide the Slot PIN when prompted.

5.7 Configuring OpenXPKI to Use Utimaco HSM

1. Configure `crypto.yaml` under `/etc/openxpki/config.d/realm.tpl` with below lines.

crypto.yaml

```
type:
 certsign: casigner
 datasafe: vault
……
token:
 casigner:
 backend: OpenXPKI::Crypto::Backend::OpenSSL
 key: "slot_33-label_Subca"
 engine: PKCS11
 engine_section: |
 engine_id = pkcs11
 dynamic_path = /usr/lib/x86_64-linux-gnu/engines-3/pkcs11.so
 MODULE_PATH = /opt/utimaco/lib/libcs_pkcs11_R3.so
 PIN = __PIN__
 init = 0
 engine_usage: 'ALWAYS'
 key_store: ENGINE
 shell: /usr/bin/openssl
 randfile: /var/openxpki/rand
 wrapper: ''
 secret: signer
 vault:
 backend: OpenXPKI::Crypto::Backend::OpenSSL
 key: "slot_33-label_Datavault"
 engine: PKCS11
 engine_section: |
 engine_id = pkcs11
 dynamic_path = /usr/lib/x86_64-linux-gnu/engines-3/pkcs11.so
 MODULE_PATH = /opt/utimaco/lib/libcs_pkcs11_R3.so
 PIN = __PIN__
 init = 0
 engine_usage: 'ALWAYS'
 key_store: ENGINE
 shell: /usr/bin/openssl
 randfile: /var/openxpki/rand
 wrapper: ''
 secret: signer
 secret:
 signer:
 label: HSM SLOT PIN
 method: literal
 value: 12345678
 cache: daemon
```

Update Slot NO., Key Label and Signer secret value according to your setup. Here Signer secret value is the HSM Slot PIN.

2. Start openxpki service.

›_ Console

```
# openxpkictl start
```

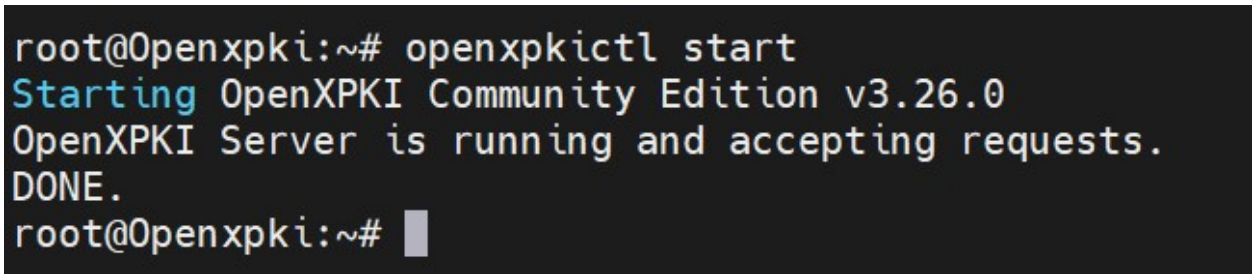

Figure 30 : Start openxpki service

3. In the process list, verify that you can see the below two processes running.

›_ Console

```
#  ps -aux | grep openxpki
```

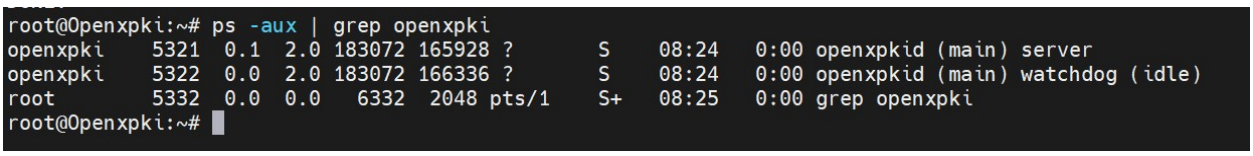

Figure 31 : openxpki process list

5.8 Importing Certificates to OpenXPKI

1. As OpenXPKI needs to be able to build the full chain for any certificate, you need to import the Root CA(s) first.

›_ Console

```
# openxpkiadm certificate import --file RootCA.pem
```

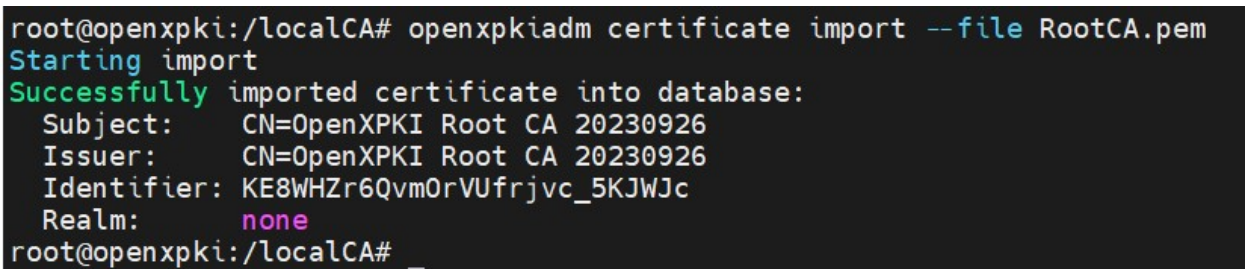

Figure 32 : Import root CA

2. Register datasafe token for the democa realm with DataVault certificate.

›_ Console

```
# openxpkiadm alias --realm democa --token datasafe --file

OpenXPKI_DataVault.crt
```

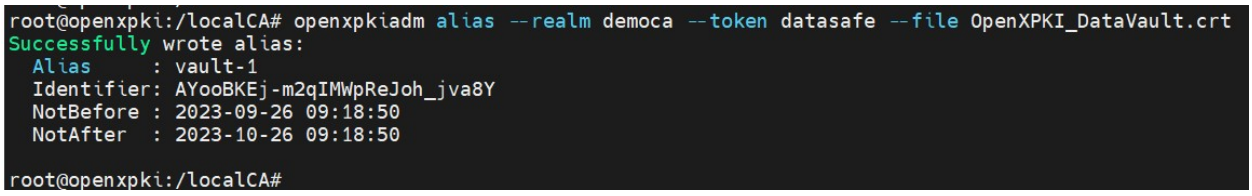

Figure 33 : Register DataVault token

3. Verify that the key_usable for DataVault token is 1.

›_ Console

```
# openxpkicli  get_token_info --arg alias=vault-1
```

```
root@openxpki:/localCA# openxpkicli  get_token_info --arg alias=vault-1
{
   "key_name" : "slot_33-label_Datavault",
   "key_secret" : 1,
   "key_store" : "ENGINE",
   "key_usable" : 1
}
root@openxpki:/localCA#
```

Figure 34 : DataVault token info

4. Register the certsign token with the Subordinate CA certificate.

›_ Console

```
# openxpkiadm alias --realm democa --token certsign --file  SubCA.pem
```

```
root@openxpki:/localCA# openxpkiadm alias --realm democa --token certsign --file  SubCA.pem
Successfully wrote alias:
  Alias     : casigner-1
  Identifier: e_YXTJkqVxOQScrGFHzDchRwMxA
  NotBefore : 2023-09-26 08:30:30
  NotAfter  : 2024-09-25 08:30:30


Token is certsign, looking for root ...
Creating alias for root ca:
  Alias     : root-1
  Identifier: KE8WHZr6QvmOrVUfrjvc_5KJWJc
  NotBefore : 2023-09-26 08:26:17
  NotAfter  : 2023-10-26 08:26:17

root@openxpki:/localCA#
```

Figure 35 : Register certsign token

5. If the import went smoothly, you should see something like this.

›_ Console

```
# openxpkiadm alias --realm democa
```

```
root@openxpki:/localCA# openxpkiadm alias --realm democa
=== functional token ===
vault (datasafe):
  Alias     : vault-1
  Identifier: AYooBKEj-m2qIMWpReJoh_jva8Y
  NotBefore : 2023-09-26 09:18:50
  NotAfter  : 2023-10-26 09:18:50

scep (scep):
  not set

casigner (certsign):
  Alias     : casigner-1
  Identifier: e_YXTJkqVxOQScrGFHzDchRwMxA
  NotBefore : 2023-09-26 08:30:30
  NotAfter  : 2024-09-25 08:30:30

ratoken (cmcra):
  not set

=== root ca ===
current root ca:
  Alias     : root-1
  Identifier: KE8WHZr6QvmOrVUfrjvc_5KJWJc
  NotBefore : 2023-09-26 08:26:17
  NotAfter  : 2023-10-26 08:26:17

upcoming root ca:
  not set

root@openxpki:/localCA#
```

Figure 36 : democa realm functional token list

6. Create a CRL.

›_ Console

```
# openxpkicmd  --realm democa crl_issuance
```

```
root@openxpki:/localCA/newcerts# openxpkicmd  --realm democa crl_issuance
Workflow created (ID: 255), State: SUCCESS
root@openxpki:/localCA/newcerts#
```

Figure 37 : CRL creation

5.9 Issuing a Test Certificate from OpenXPKI Web UI

1. Navigate your browser to http://<openxpki_server_ip>:8080.
2. Log in as authenticated user.

Figure 38 : Home page of openxpki user

3. Go to **Request**, select **Request new certificate** and select any certificate profile.

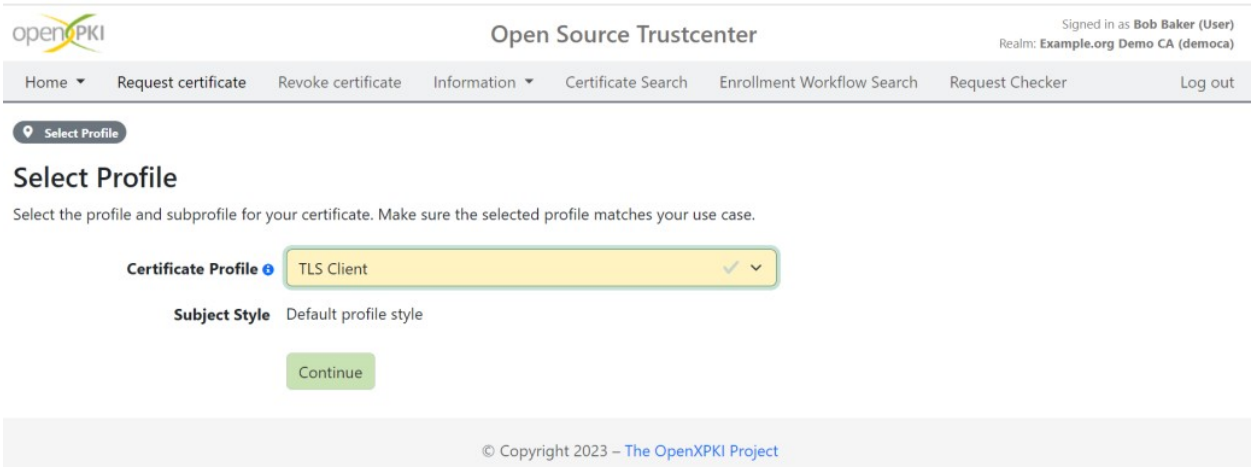

Figure 39 : Request for new certificate

4. Click **Continue**.
5. Click on **Generate Key on PKI**.

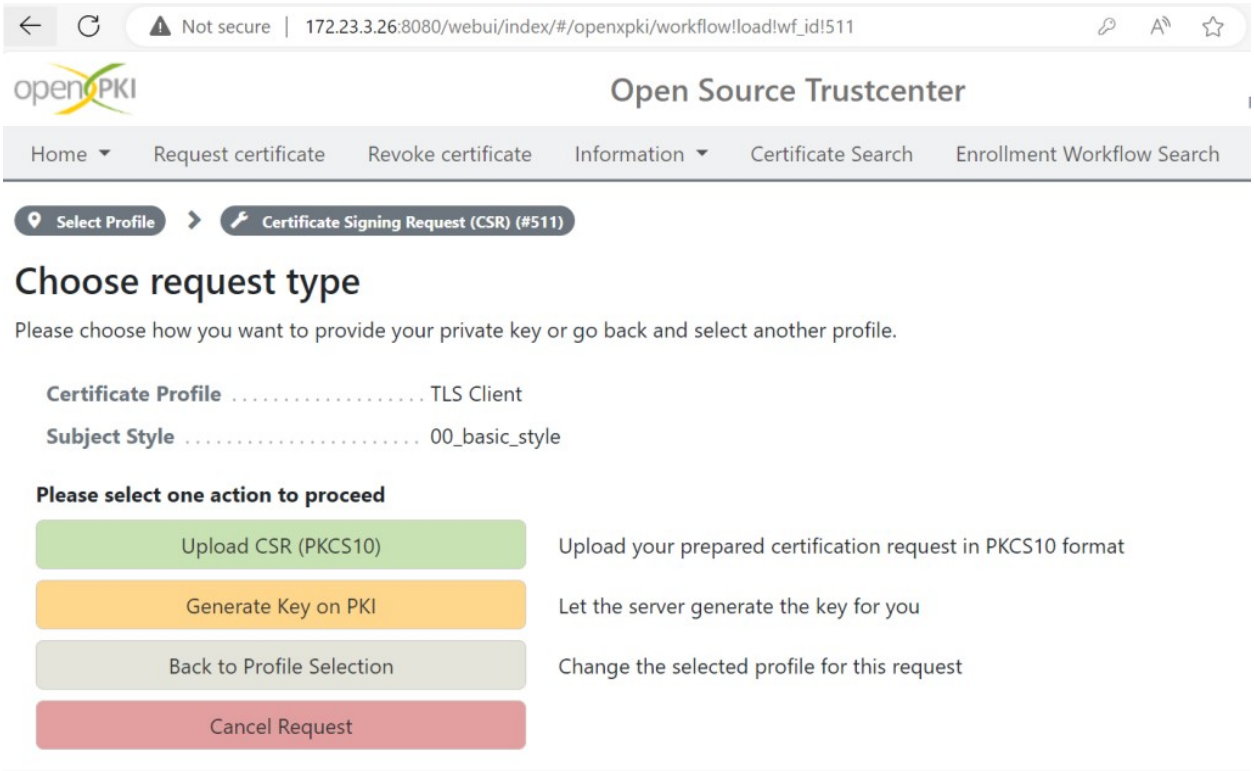

Figure 40 : Generating key on PKI

6. Select **Key Algorithm** and **Key length** then click on **Continue**.

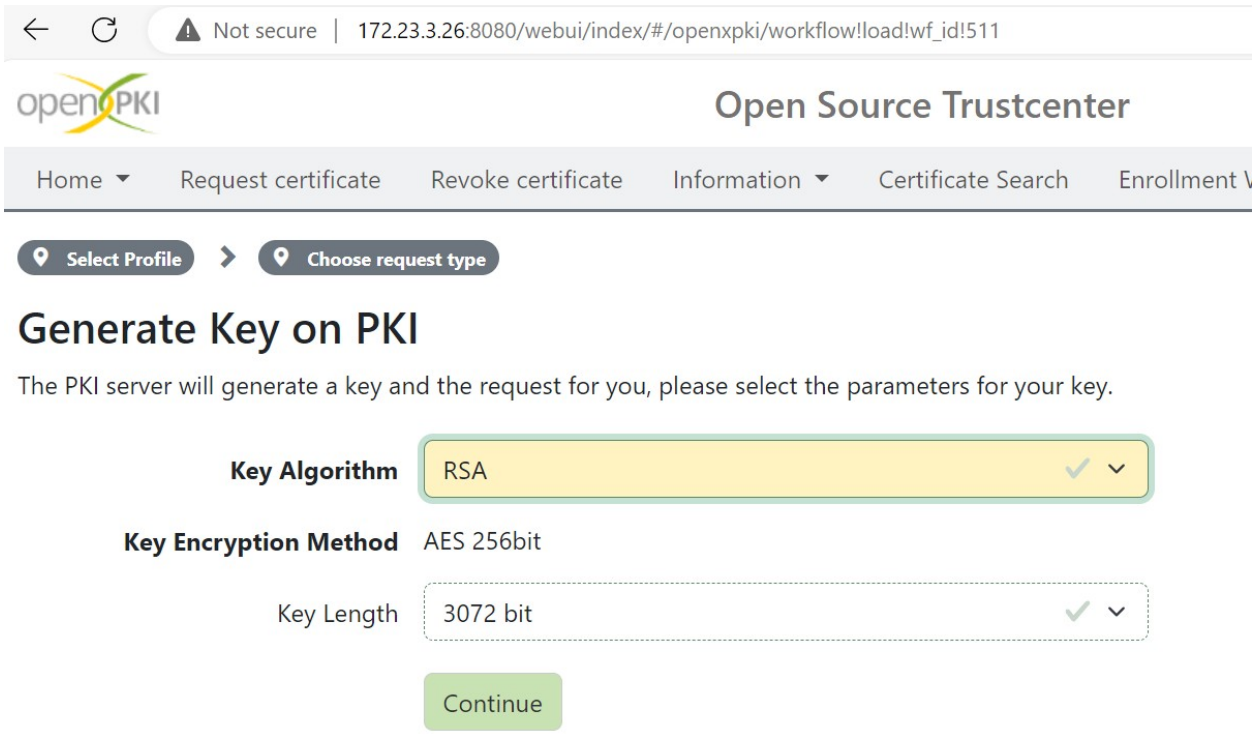

Figure 41 : Select key algorithm and length

7. Enter the **Hostname** and **Application Name** of your machine and click on **Continue** tab.

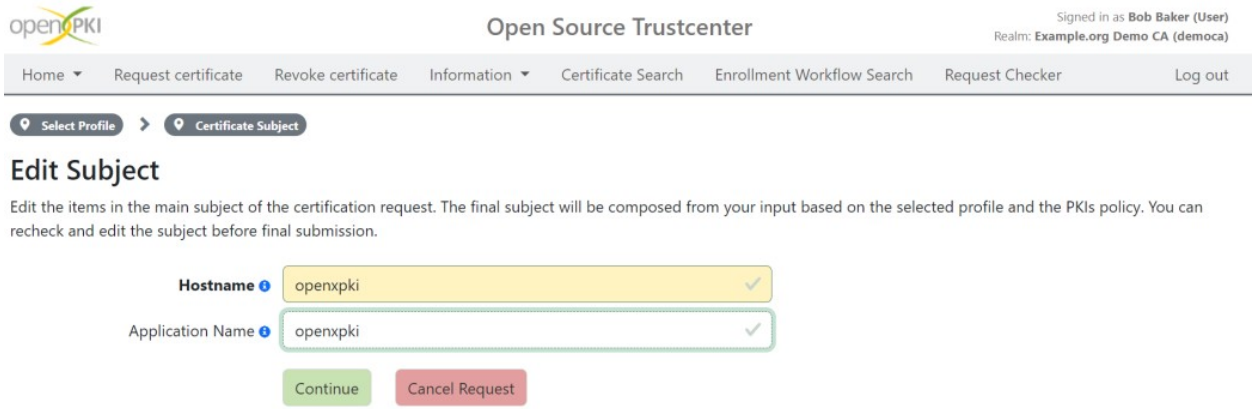

Figure 42 : Update host and application details

8. Provide the **Email address** and other information and click on **Continue** tab.

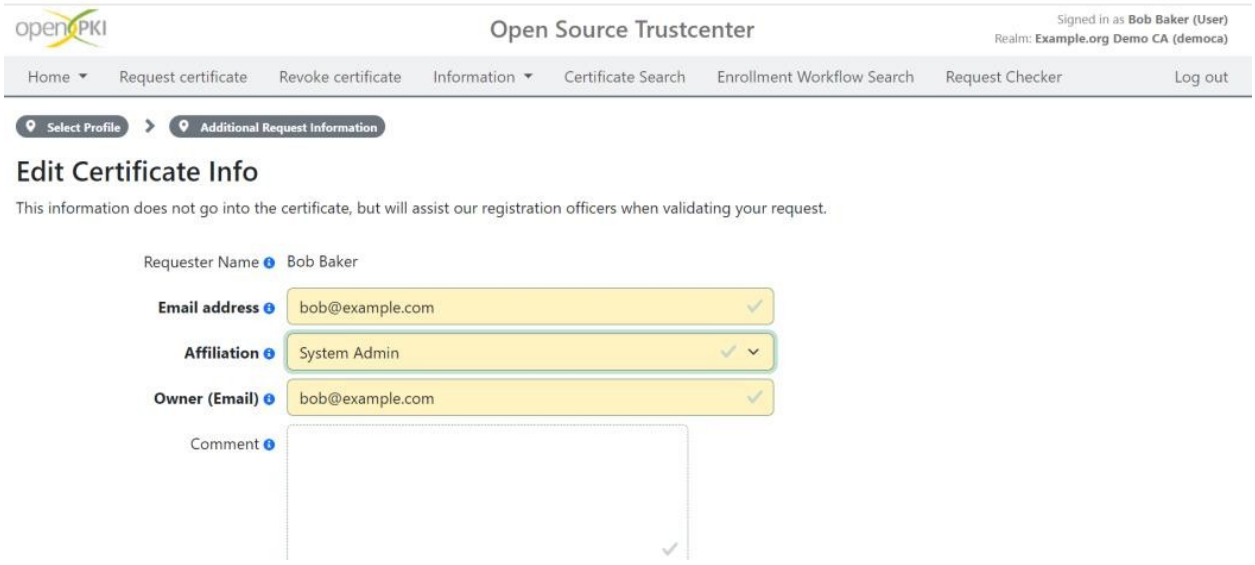

Figure 43 : Update other details

9. Click on **Submit request**.

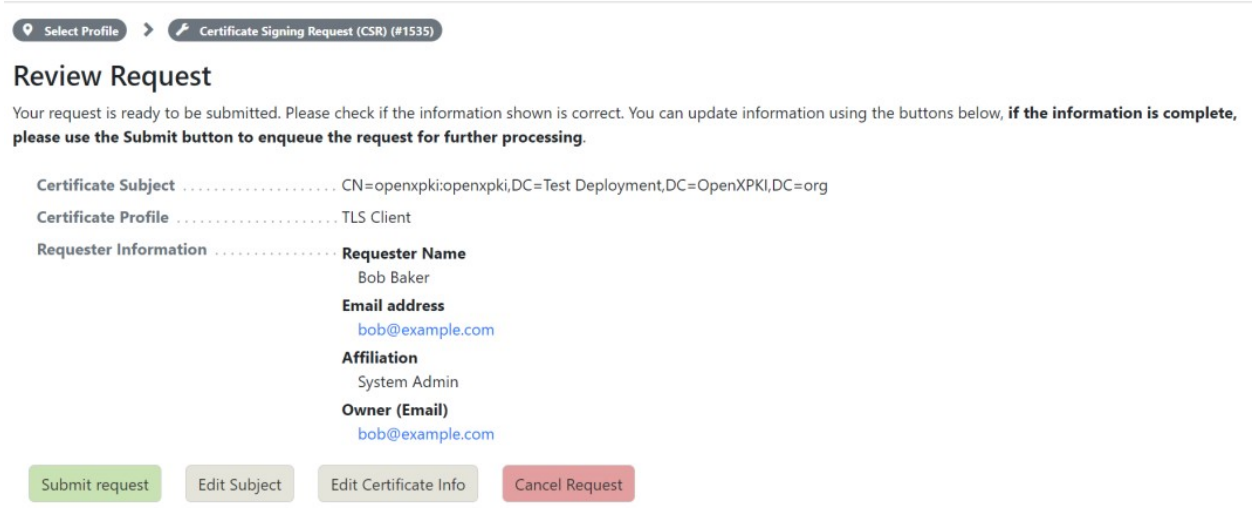

Figure 44 : Submit Request

10. Enter your Key **Password**.

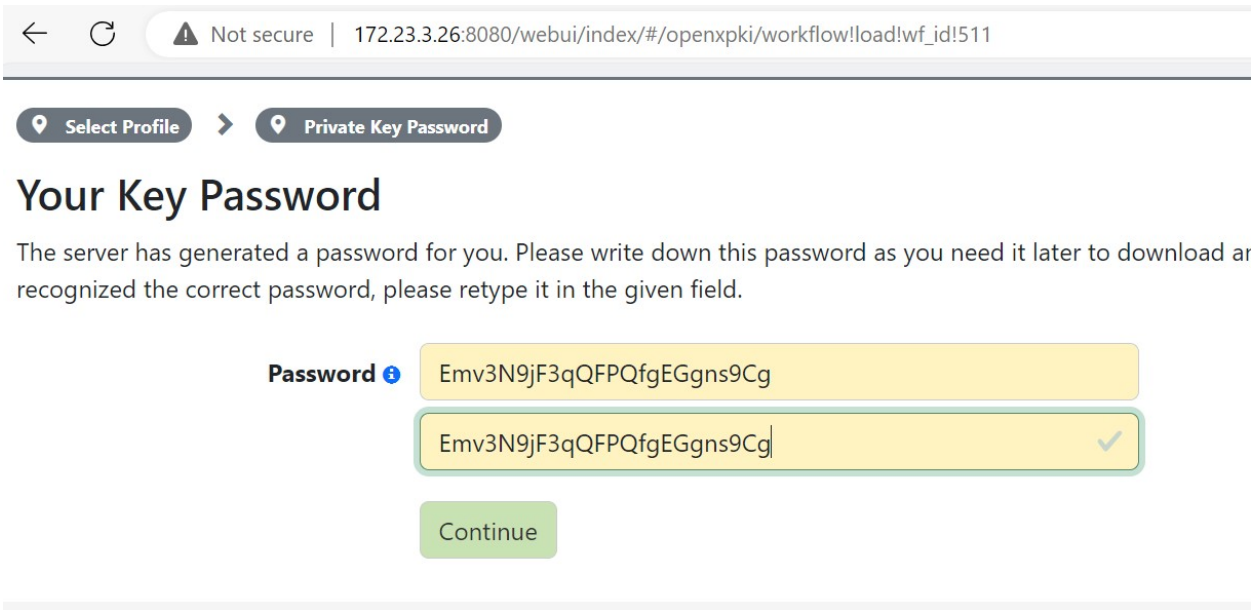

Figure 45 : Set Key Password

11. Click on **Continue**.
12. Request awaits for the approval.

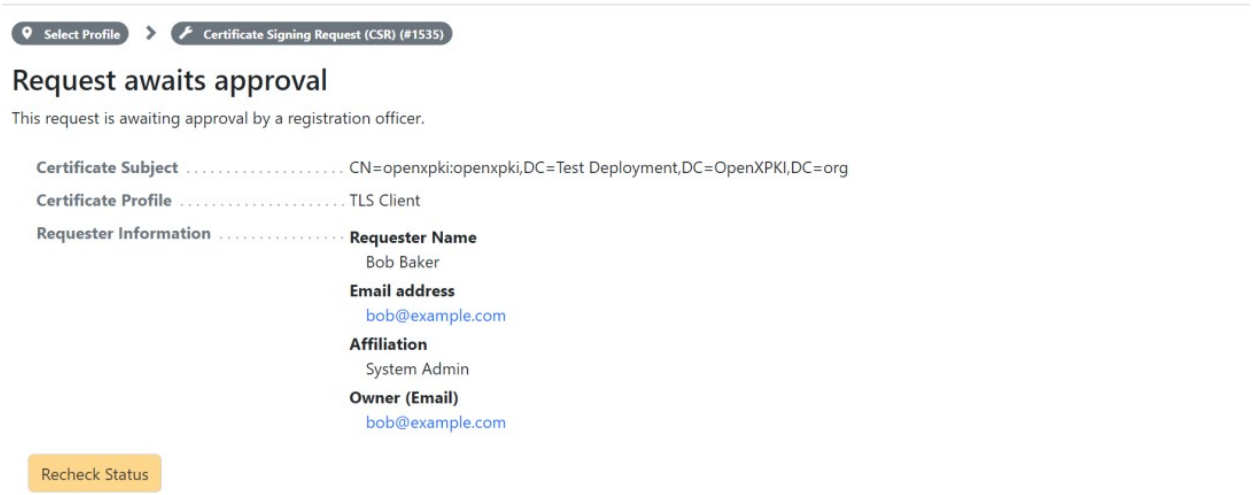

Figure 46 : Request for approval

13. Log out and re-login as user who has permission to approve and issue certificate.
14. Select **Home** and click on **My Tasks**, there should be a table with one request pending.
15. Select your Request by clicking the line.

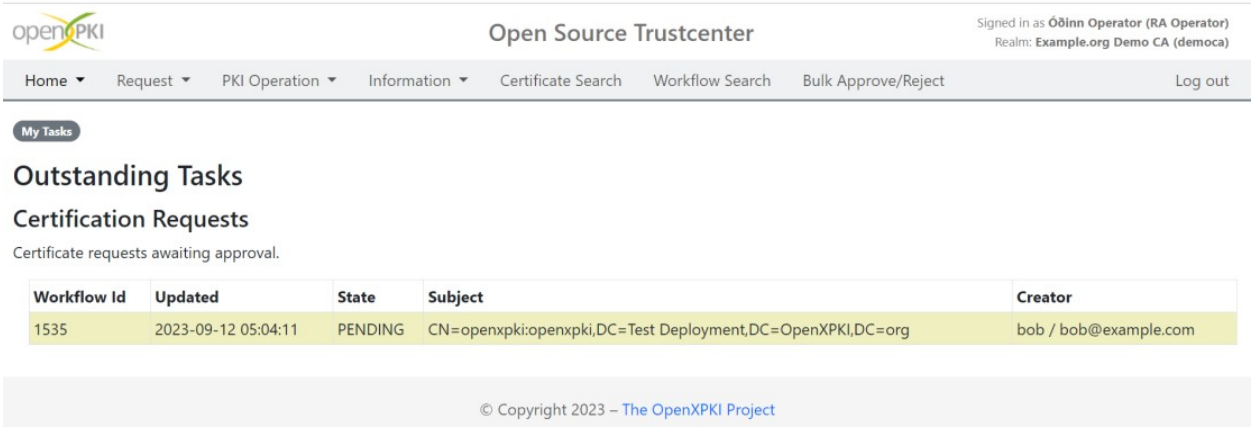

Figure 47 : My tasks page

16. Click the **Approve Request** button.

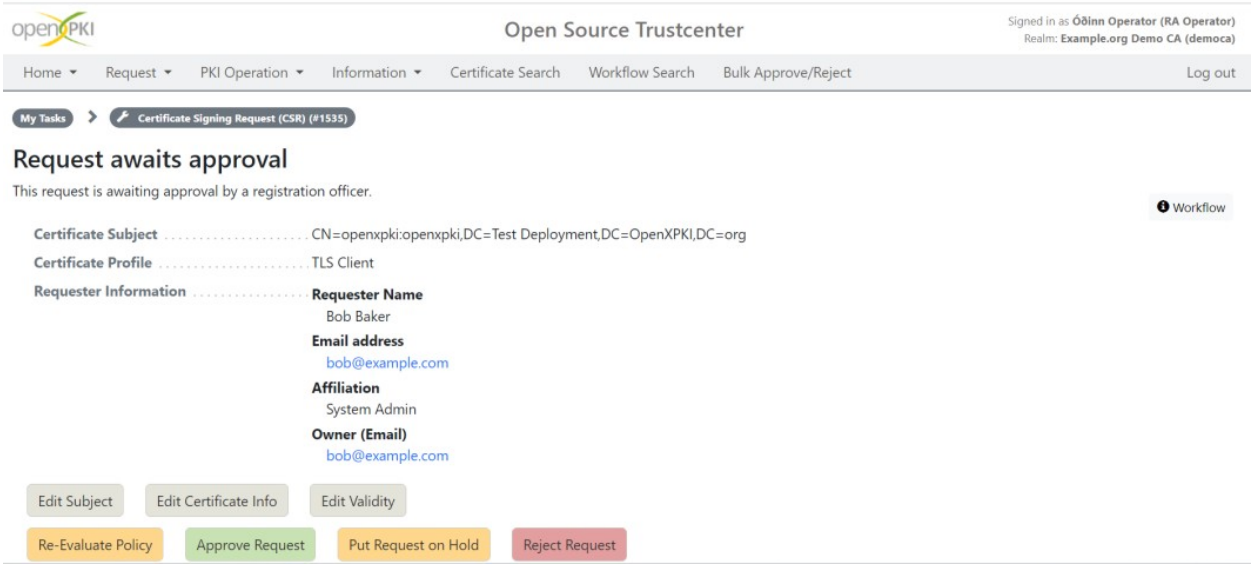

Figure 48 : Approving request

17. After few seconds, your first certificate is issued.

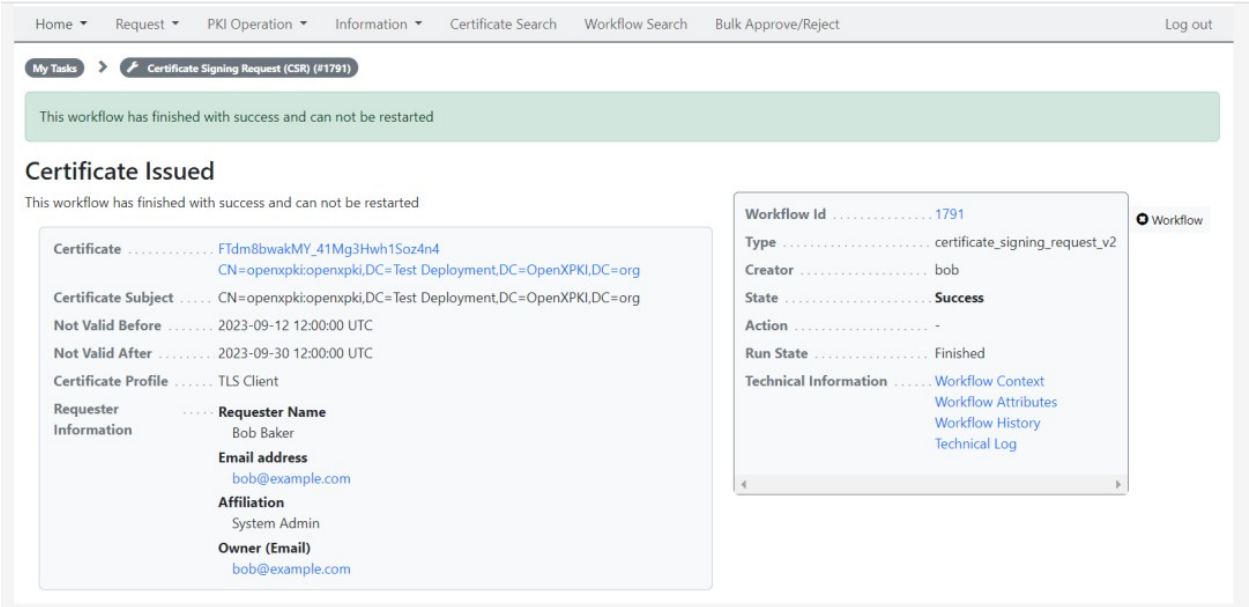

Figure 49 : Certificate Issued

18. Log out and log in with your requestor username. Open **My Certificates**. Now you can download the issued certificate.

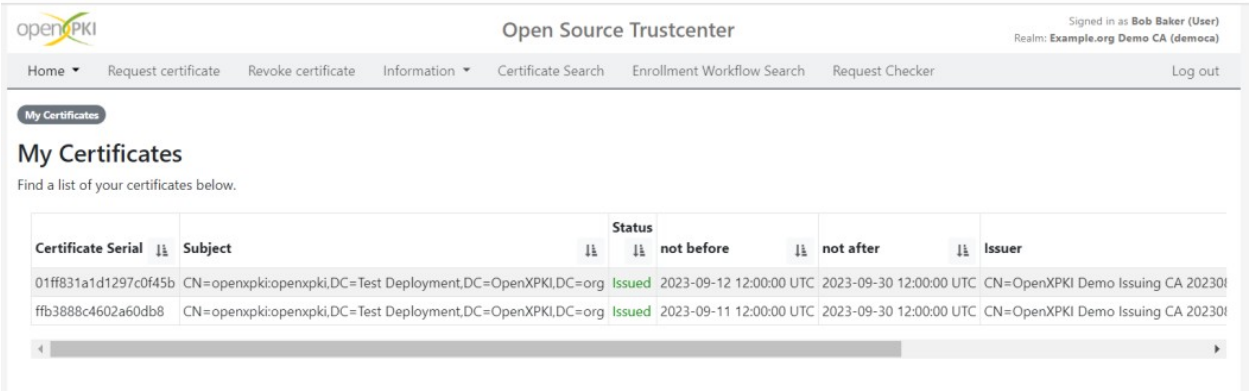

Figure 50 : Issued certificate options

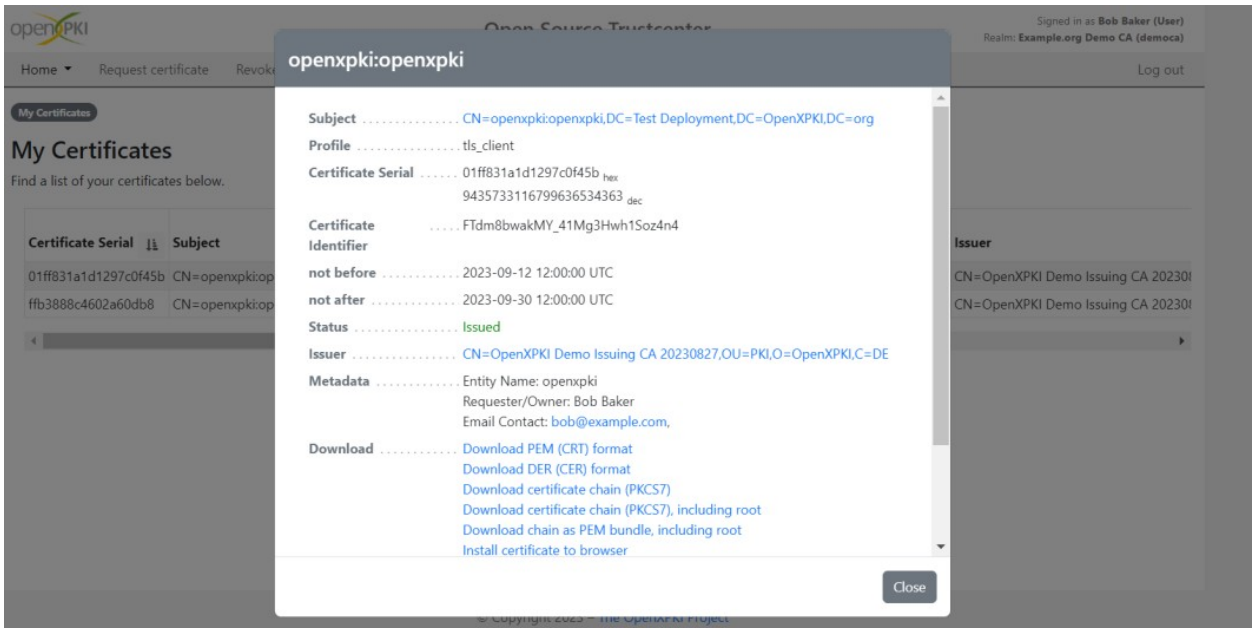

Figure 51 : Issued certificate options

This completes the integration for OpenXPKI with Utimaco SecurityServer.

6 Troubleshooting

Error	Diagnosis
Unable to access openxpki page with error: 18n_openxpki_ui_application_error	1.dpkg-reconfigure locales install en_US.utf8 2. update-locale LANG=en_US.UTF-8
configure: error: libcrypto >= 0.9.8 is required	1.apt-get install libssl-dev 2.apt-get install pkg-config
CA certificate and CA private key do not match 808BDB6E067F0000:error:05800074:x509 certificate routines:X509_check_private_key:key values mismatch:../crypto/x509/x509_cmp.c:405:	Verify the HSM slot no and key label is correct in file /etc/openxpki/config.d/realm.tpl/crypto.yaml

Table 6: List of errors and their diagnoses

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 References

Reference	Title/Company	Document No.
[CSADMIN]	CryptoServer – csadm Manual/Utimaco IS GmbH	2009-0003
[CSTrSh]	CryptoServer Troubleshooting/Utimaco IS GmbH	M011-0008-en
[CSADMIN2]	CryptoServer_csadm_Manual_Systemadministrators.pdf	2009-0003
[CSP11Tool2]	CryptoServer_p11tool2_Manual.pdf	2012-0004
[CSPKCSM]	CryptoServer - PKCS#11 P11CAT Manual	M013-0001-en
[CSLAN5]	CryptoServerLAN_Manual_Systemadministrators.pdf	2018-0004

Table 7: References

9 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.