

# Application Notes

## Create Self-signed certificate for OPC UA connection

---



Exported from Kemro X Wiki

### Disclaimer

The recommendations for action on which these Application Notes are based have been developed within the framework of tests under the ambient conditions specified in the operating instructions. The user is responsible for compliance with and verification of these environmental conditions in the specific application.

These Application Notes are intended for qualified personnel who commission and maintain drive and automation components. According to IEC 60364 or CENELEC HD 384, qualified personnel are persons who have the appropriate qualifications and are familiar with the installation, assembly, commissioning and operation of KEBA products (electrical devices) and who are familiar with all accident prevention regulations, directives and laws applicable at the place of use.

The safety instructions contained in the device documentation of the respective device must be observed.

The screenshots shown in these Application Notes are only examples to illustrate the individual steps.

Please note that KEBA products may contain software that is licensed as Open Source Software (OSS) or Free Software (FOSS). The license conditions of the OSS and / or FOSS contained / used in the products are available on the DevAdmin Service webpage on the controller. These must be complied with.

All information is subject to change at any time. Liability for correctness and completeness is excluded.

# Create Self-signed certificate for OPC UA connection

RELEASED EXTERNAL

## Version info

Version	Kemro X
Product /Component name	KeTop devices
Product /Component version	Win10 XCA 2.4.0

## Additional required data

- Link to website: [X - Certificate and Key management](#)
- [Example of an OpcUaltf.cfg](#)

- 
- 1 [Description](#)
  - 2 [Creating self-signed certificate using Windows](#)
    - 2.1 [In "X - Certificate and Key management" \(XCA\), create a new self-signed certificate.](#)
  - 3 [Creating self-signed certificate using Linux \(Ubuntu\)](#)
- 

## Description

This document helps you to create a self signed certificate. You'll need this when you are using a secured and encrypted connection.

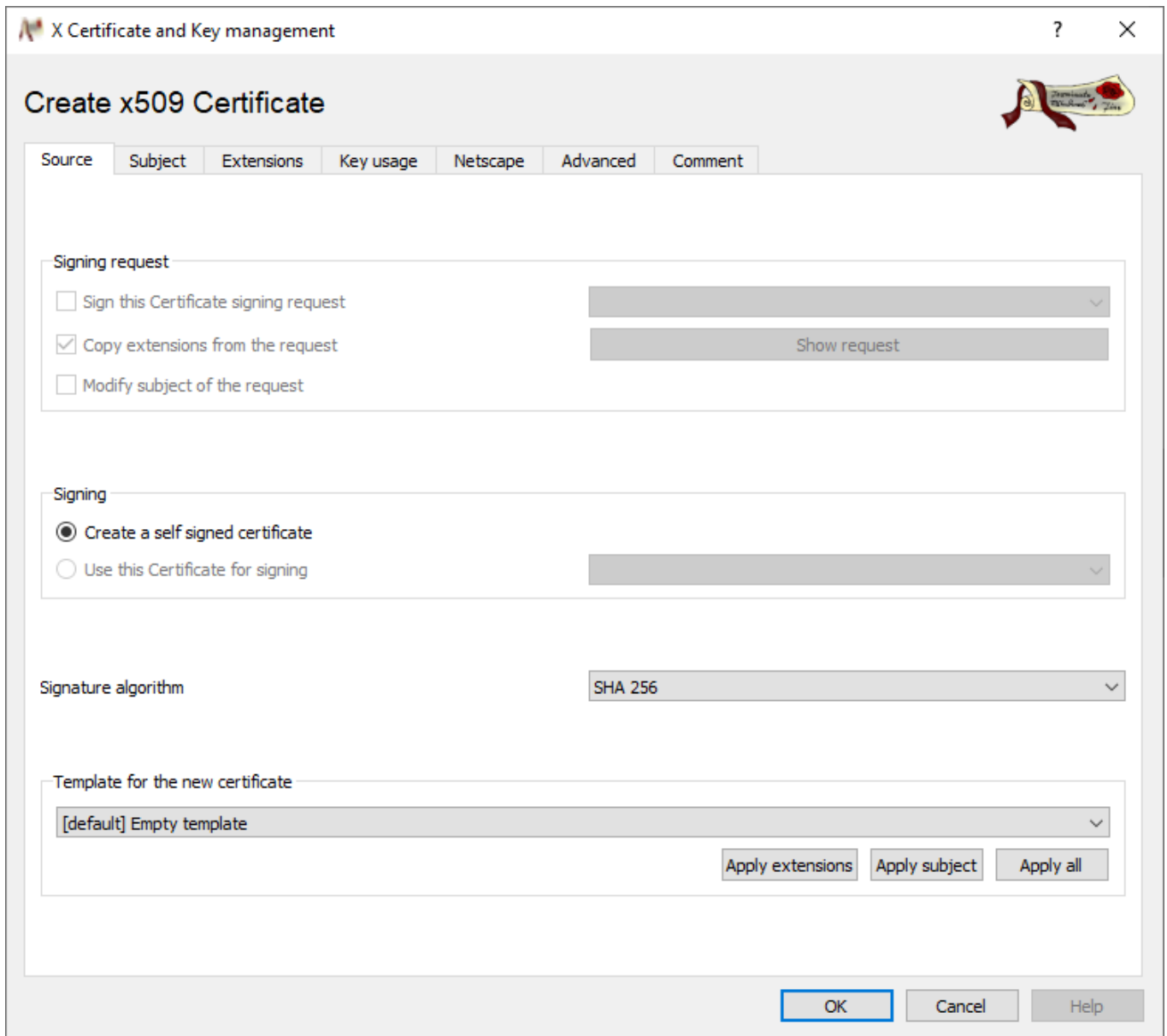
---

## Creating self-signed certificate using Windows

In "X - Certificate and Key management" (XCA), create a new self-signed certificate.

- Open (or create) a database under the menu item **"File"**.
- In the **"Certificates"** tab, click the **"New Certificate"** button:





- Switch to the **“Subject”** tab and fill out the fields like shown below.

## Create x509 Certificate



Source Subject Extensions Key usage Netscape Advanced Comment

Internal Name

Distinguished name

countryName  organizationalUnitName

stateOrProvinceName  commonName

localityName  emailAddress

organizationName

Type	Content

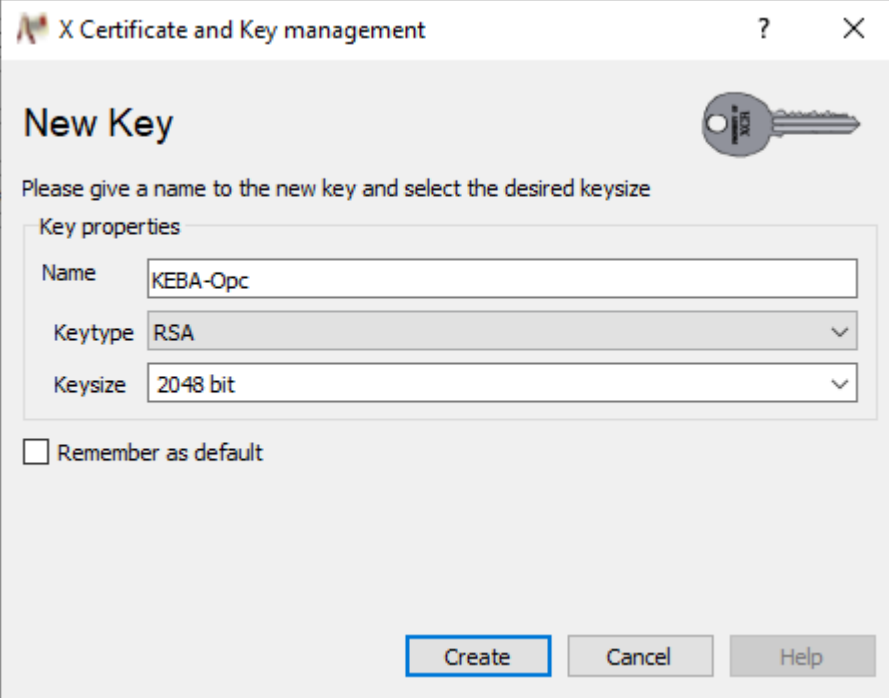
Add  
Delete

Private key

Used keys too

OK Cancel Help

- Click the **“Generate a new key”** button. This generates a unique private key for this certificate. Use Keytype **“RSA”** and Keysitze **“2048 bit”**



The screenshot shows a window titled "X Certificate and Key management" with a "New Key" dialog box. The dialog box has a key icon and the text "Please give a name to the new key and select the desired keysize". Under "Key properties", there are three fields: "Name" with the value "KEBA-Opc", "Keytype" with the value "RSA", and "Keysize" with the value "2048 bit". There is a checkbox labeled "Remember as default" which is unchecked. At the bottom, there are three buttons: "Create", "Cancel", and "Help".

- In the “**Extensions**” tab, set the type to **CA** or **End Entity**. The example shows a validation of 1 year. If more or less is needed change the period accordingly. Then insert the URI into the field “**X509v3 Subject Alternative Name**”. This field must not be empty – this is a requirement of the OPC UA specification. The URI must exactly be this: **urn:KEBA-PC:Keba:Opcua1f**

X Certificate and Key management

### Create x509 Certificate

Source Subject Extensions Key usage Netscape Advanced Comment

**X509v3 Basic Constraints**

Type:   Critical

Path length:

**Key identifier**

X509v3 Subject Key Identifier  
 X509v3 Authority Key Identifier

**Validity**

Not before:   
 Not after:

**Time range**

Midnight  Local time  No well-defined expiration

X509v3 Subject Alternative Name

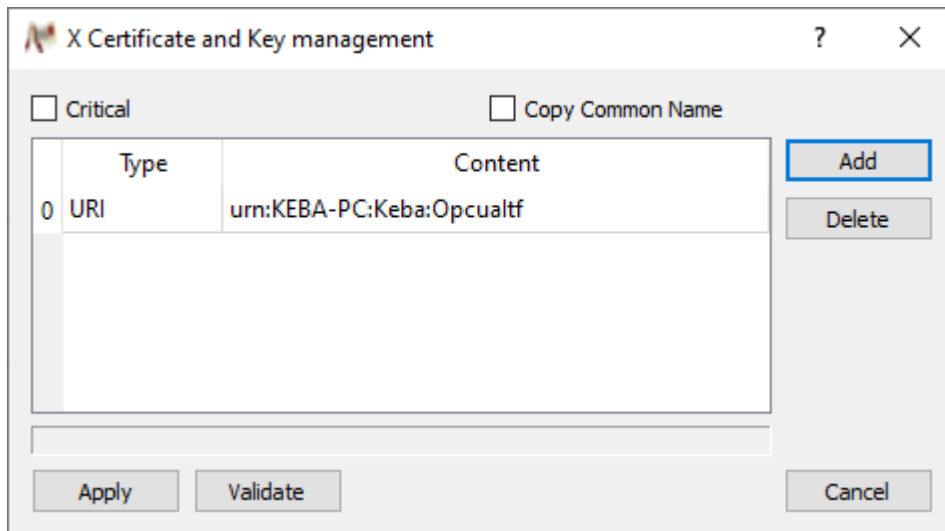
X509v3 Issuer Alternative Name

X509v3 CRL Distribution Points

Authority Information Access

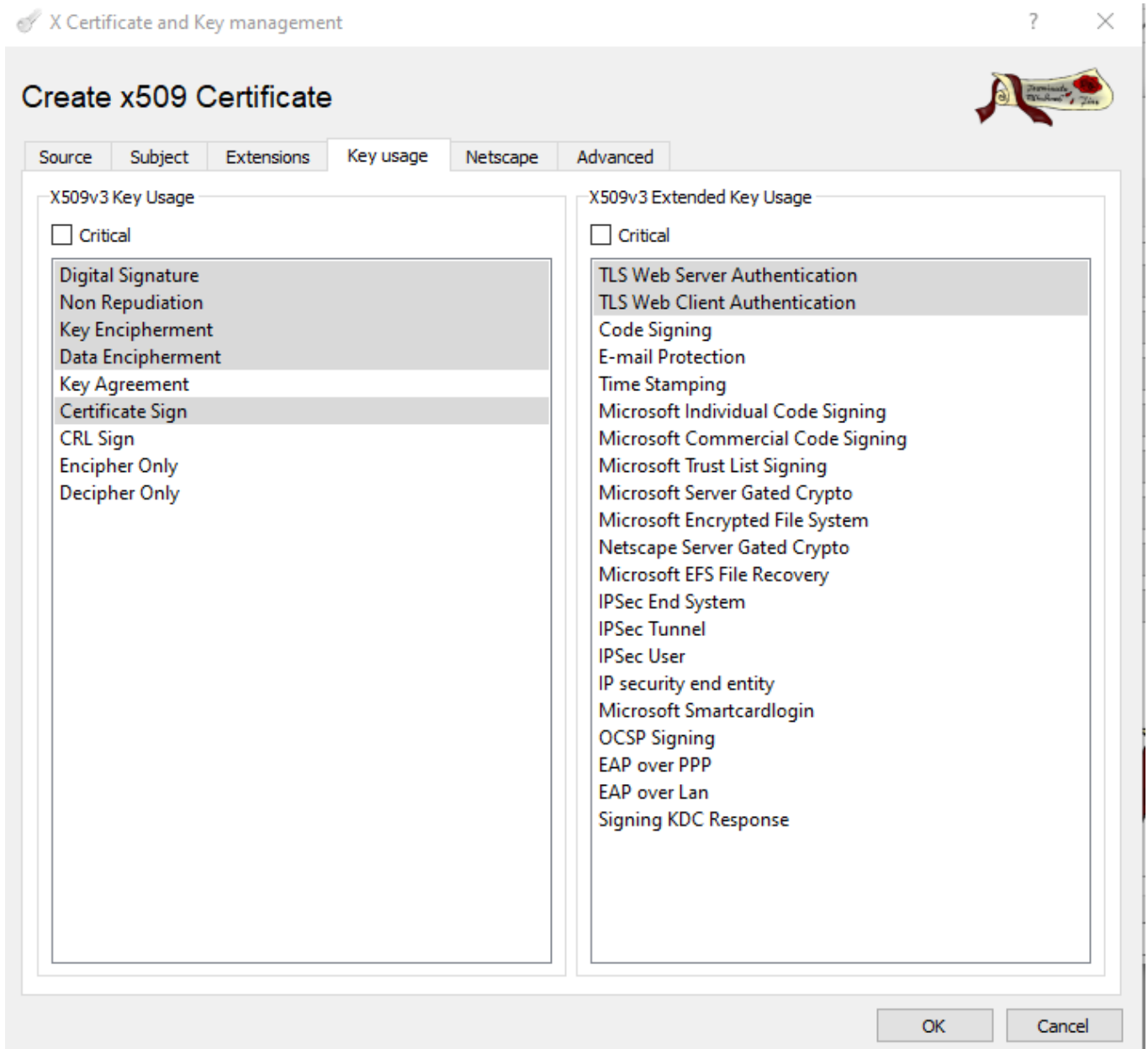
OCSP Must Staple

- This is how it looks like when you press the "Edit" Button next to the input field "X509v3 Subject Alternative Name".

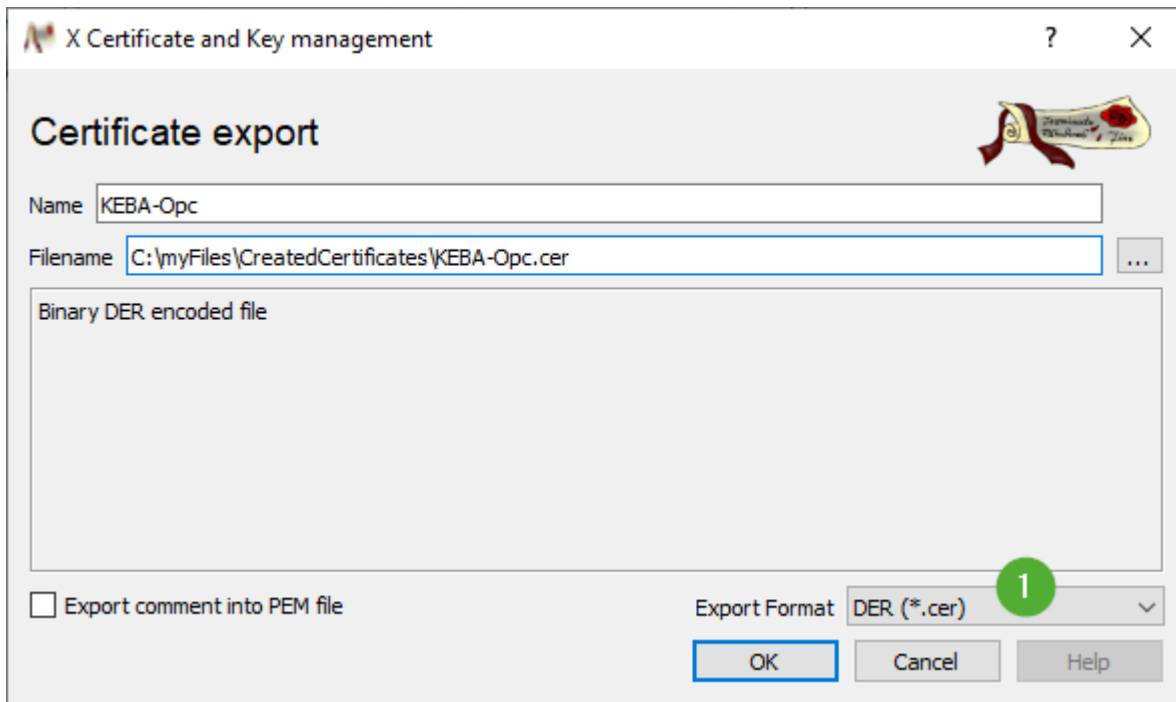


- In the **"Key usage"** tab you have to select the need keys. Select **"Digital Signature"**, **"Non Repudiation"**, **"Key Encipherment"**, **"Data Encipherment"** and **"Certificate Sign"** on the left side. On the right side **"TLS Web Server Authentication"** and **"TLS Web Client Authentication"**.

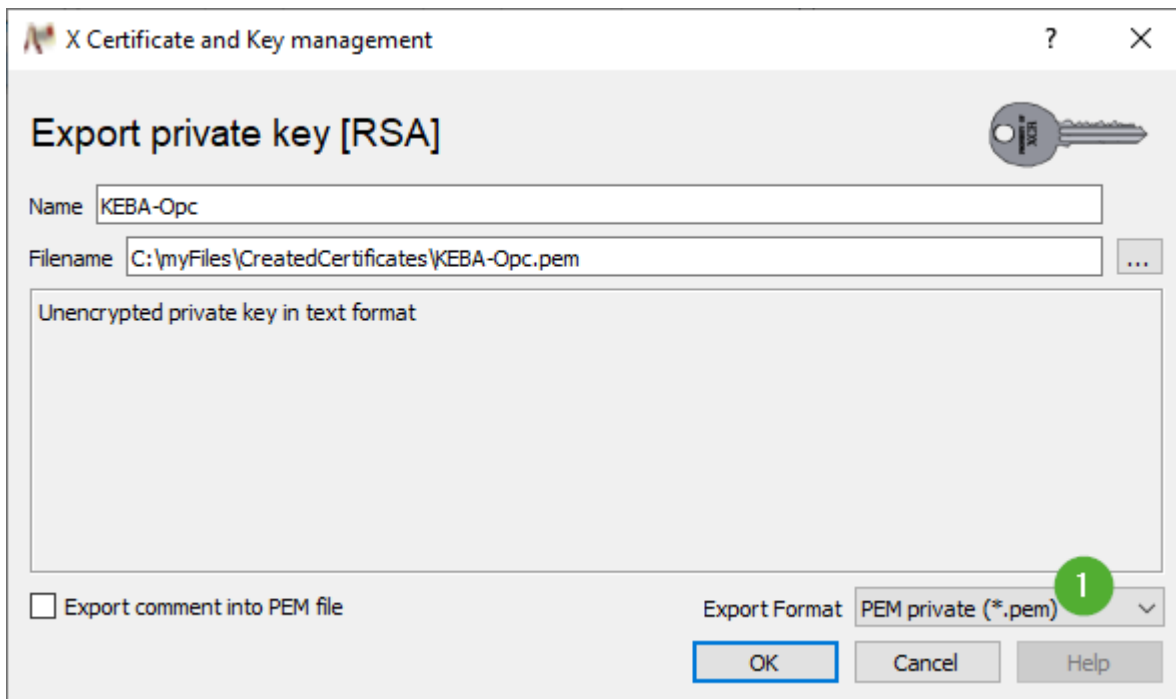




- Click "**OK**" to create the certificate.
- After the certificate is created, select the "**Certificates**" tab in the main XCA window and click "**OK**" to export. Export the certificate in the format "DER (\*.cer)":



- Then, also export the private key. Select the "**Private Keys**" tab in the main XCA window and click "**OK**" to export. Export the private key in the format "PEM private (\*.pem)":



- Now, the creation of the certificate (**KEBA-Opc.cer**) and the private key (**KEBA-Opc.pem**) is finished. Copy the two files now to a folder on the KeTop (f.e. "C:\Cert\)")

## Creating self-signed certificate using Linux (Ubuntu)

- Create following folder structure on you Linux machine:

```
`-- opc
  |-- tools
  |-- certs
  `-- created
```

or open Terminal and run this code:

- `mkdir -p opc/{tools,created}/certs`

- Go to Github: <https://github.com/open62541/open62541/blob/master/tools/certs/>
- Copy the folder "certs" with the two files (create\_self-signed.py and localhost.cnf) local in any folder (here in this example it is: /opc/tools/certs/)
- Open Terminal (or any other command line tool)
- Run code:

```
cd opc/tools/certs
```

- Run code:  

```
python3 ./create_self-signed.py -u "urn:KEBA-PC:Keba:OpcuaItf" -c "KEBA-Opc" -k 2048 ~/opc
```
- Now, the creation of the certificate (**KEBA-Opc\_cert.der**) and the private key (**KEBA-Opc\_key.der**) is finished. Copy the two files now to a folder on the KeTop (f.e. "C:\Cert")

## Known Issues

- None