T5 protocol over TLS

straton user guide – Rev. 1

sales@straton-plc.com

CP_R31_006_H

straton



STRATON AUTOMATION, All Rights Reserved

The information contained in this document is the property of STRATON AUTOMATION. The distribution and/or reproduction of all or part of this document in any form whatsoever is authorized only with the written authorization of STRATON AUTOMATION. The technical data are used only for the description of the product and do not constitute a guarantee of quality in the legal sense of the term. We reserve the right to make technical changes.

Content

1. GLOBAL ARCHITECTURE 4

2. USE XCA SOFTWARE TO CREATE CERTIFICATES 5

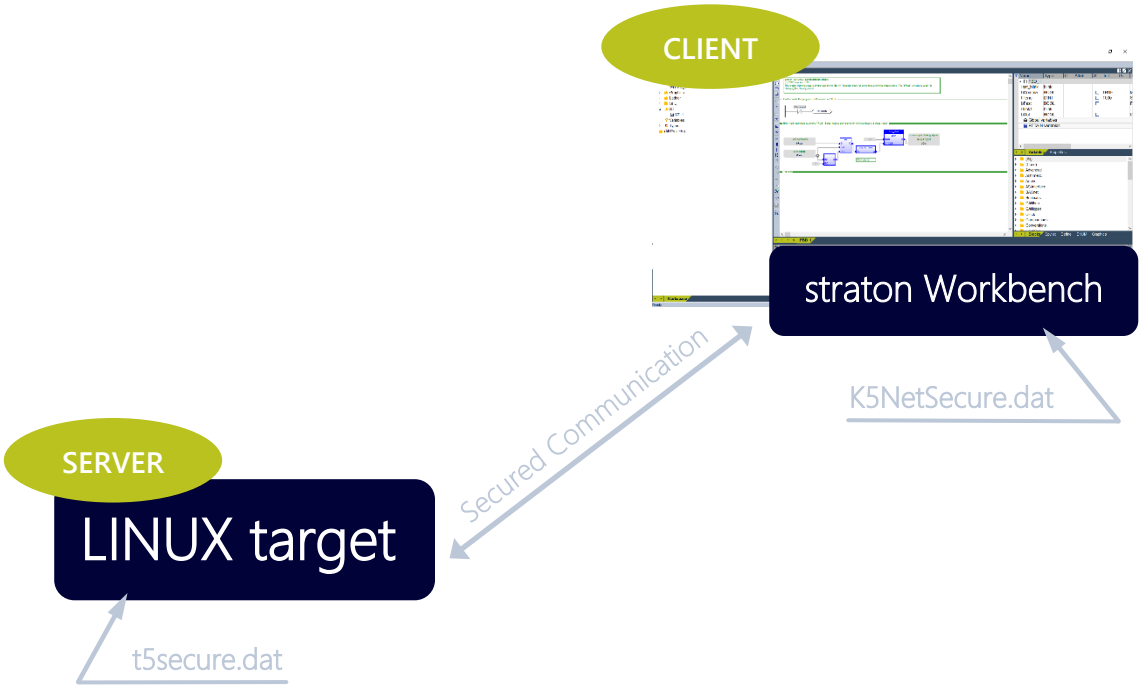
2.1. Create (self-signed) Certificate Authority (CA) 5
2.2. Create CLIENT certificate 6
2.3. Create SERVER certificate 6
2.4. Example of storage folders tree structure 7
2.5. Trustlist (*.0) 7
2.6. Private certificates (*.pem) 8

3. CREATE CONFIGURATION FILES (*.DAT) 9

3.1. General Parameters and Values 9
3.2. Configuration files – Samples 10

4. EXAMPLE OF IMPLEMENTATION IN STRATON 11

1. Global Architecture



2. Use XCA Software to create certificates

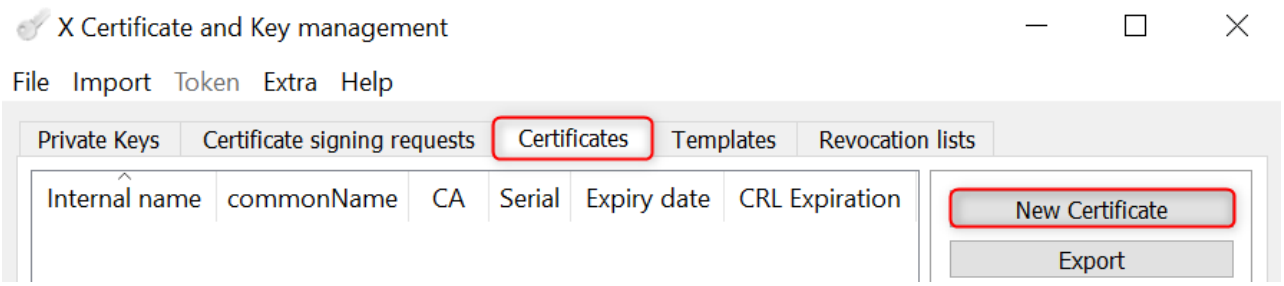
This application is intended for creating and managing X.509 certificate, start your own PKI and create all kinds of private keys, certificates and/or CRLs (Certificate Revocation List). You can download the Software and find more information on the official website: https://hohnstaedt.de/xca/index.php

2.1. Create (self-signed) Certificate Authority (CA)

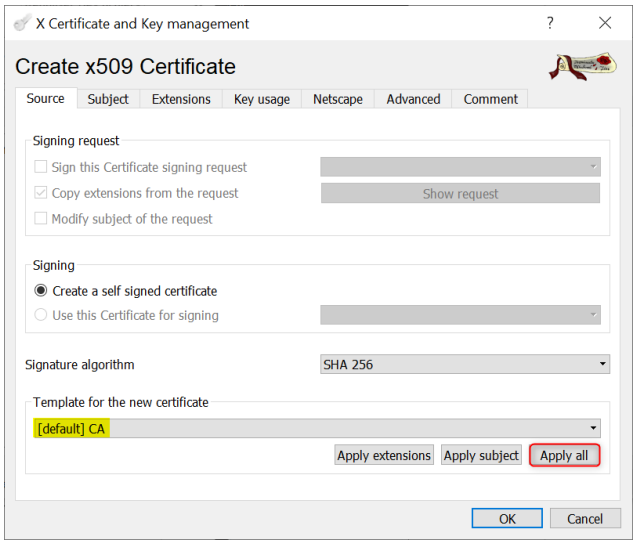
To create certificates, go to the "**Certificates**" tab then click on "**New Certificate**"

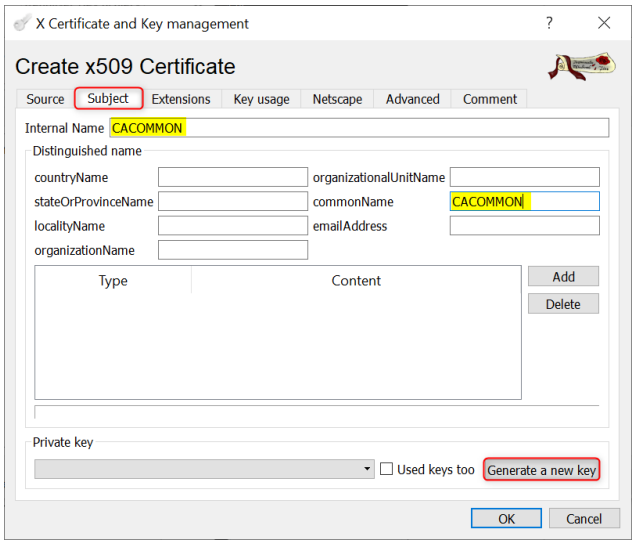
The **CA** (**C**ertificate **A**uthority) will be used to sign the other certificates.

Note that the one created here is only for test purposes.



- In the "**Source**" tab, choose the template "**[default] CA**" then click on "**Apply all**"
- After that go to the "**Subject**" tab, choose an *Internal Name* and a *commonName* (here we will use "**CACOMMON**"), click on "**Generate a new key**" then select "**Create**"

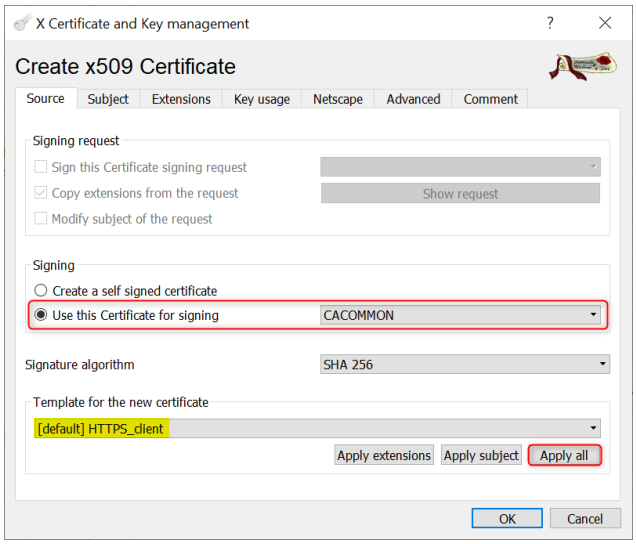


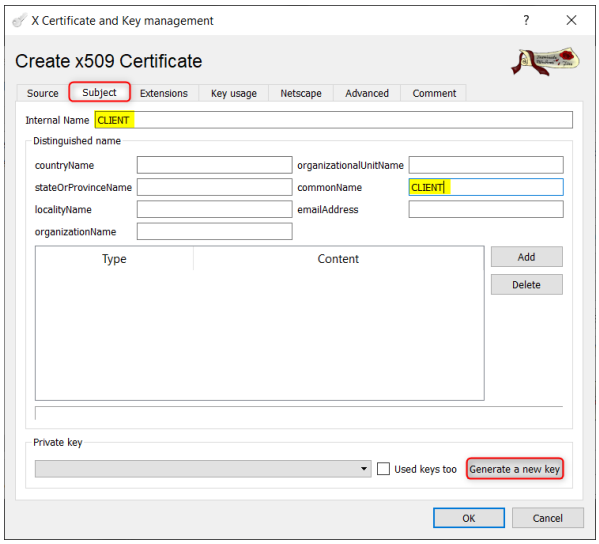


2.2. Create CLIENT certificate

Now we will create the certificate for the **CLIENT** :

- Click on "**New Certificate**", select "**Use this Certificate for signing**" to use CACOMMON as signing authority, choose the template "**[default] HTTPS_client**" then click on "**Apply all**".
- Same as for the CA, go to the "**Subject**" tab, choose an *Internal Name* and a *commonName* (here we will use "**CLIENT**"), click on "**Generate a new key**" then select "**Create**"



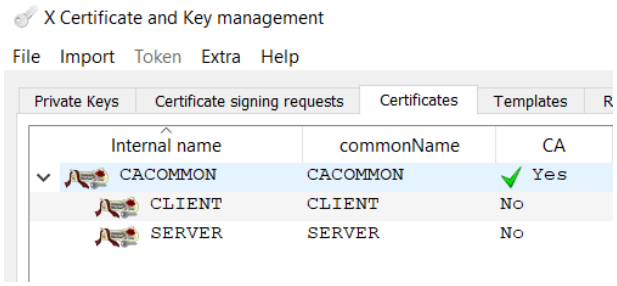


2.3. Create SERVER certificate

Same principle for the **SERVER** :

- Click on "**New Certificate**", select "**Use this Certificate for signing**" to use CACOMMON as signing authority, choose the template "**[default] HTTPS_server**" then click on "**Apply all**".
- Go to the "**Subject**" tab, choose an *Internal Name* and a *commonName* (here we will use "**SERVER**"), click on "**Generate a new key**" then select "**Create**"

After all these steps you should have something like this :



2.4. Example of storage folders tree structure

For the next steps you will need to create the following tree structure :

C: *(for example)*\PKI\T5TLSC\

- CA
- PRIVATE

C: *(for example)*\PKI\T5TLSS\

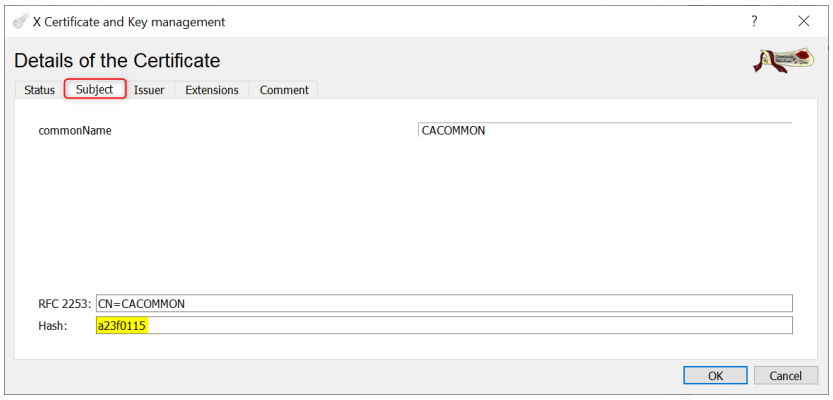
- CA
- PRIVATE

/!\ do not change the name of sub-folders "**CA**" and "**PRIVATE**"

2.5. Trustlist (*.0)

The **Trustlist** will be used (as its name suggests) to trust/or not our certificates.

- Double-click on the CA certificate (CACOMMON in this example), go to the "**Subject**" tab then copy the "**Hash**". If you've followed our example it must be "**a23f0115**".



- Right-click on the "**CACOMMON**" certificate, click on "**Export**" and select "**File**"
- Change the *Filename* path to match with the storage folders architecture we created previously and put it in one of the "**CA**" folder
- Change the name of the registered file to have : <HASH>**.0** *(or with our example : C:\PKI\T5TLSC\CA\a23f0115.0)*

/!\ the *Export Format* must be "**PEM (*.crt)**" but the file extension must be ***.0**

As it is the same Certificate Authority (CA), the trustlist will be the same for the CLIENT and the SERVER.

If you'll have registered the previous "Trustlist" in the CLIENT's CA folder, redo the same thing to put a second one into the SERVER's one *(example path : C:\PKI\T5TLSS\CA\a23f0115.0)* or do a copy/paste from the CLIENT to the SERVER.

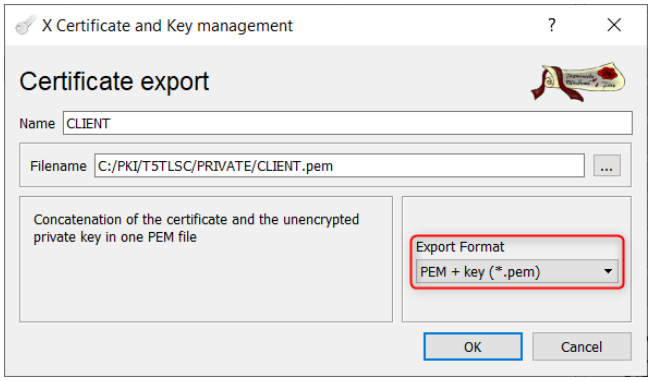
2.6. Private certificates (*.pem)

Now we will export CLIENT's and SERVER's private certificates.

Let's start with the CLIENT :

- Right-click on the "CLIENT" certificate, click on "Export" and select "File"
- Change the *Filename* to put it in the CLIENT's PRIVATE folder
- Change the *Export Format* to "PEM + key (*.pem)"

Note that, unlike the Trustlist, the certificate file's name does not matter.



Do the same with the SERVER :

- Right-click on the "SERVER" certificate, click on "Export" and select "File"
- Change the *Filename* to put it in the SERVER's PRIVATE folder
- Change the *Export Format* to "PEM + key (*.pem)"

3. Create configuration files (*.dat)

The encryption/authentication will works using two configuration files (K5NetSecure.dat & t5secure.dat).

- K5NetSecure.dat *(configuration file used by the Workbench)* : It must be located in the "DATA" folder which could be founded (by default) here

  => C:\Users\Public\Documents\Straton\STRATON XX.X.X.XXXXX

- t5secure.dat *(configuration file used by the target)* : It must be located in the same folder as the Runtime binary

3.1. General Parameters and Values

Each *.dat file uses following parameters. Some are specific to one of them and others have default (hardcoded) values if not defined in the file.

Parameters	Key Words	Values
Private key file	KEY_CERT	*STRING* - Certificate filename (without path) */!\ Mandatory /!*
Public certificate file	PUB_CERT	*STRING* – Public certificate filename (with path) *Hardcoded : EMPTY – use the same as "KEY_CERT"*
Min TLS version allowed	TLS_MIN	*(TLS_NONE)* 0 *(TLS_11)* 1 *(TLS_12)* 2 *(Hardcoded/Default)* *(TLS_13)* 3
Max TLS version allowed	TLS_MAX	*(TLS_NONE)* 0 *(TLS_11)* 1 *(TLS_12)* 2 *(TLS_13)* 3 *(Hardcoded/Default)*
Certificates path	PKI_PATH	*STRING* - Path (to /CA & /PRIVATE folders)
Permissive ciphers	CIPHERS	*STRING* - Enumeration separate by colons (OpenSSL format) *Hardcoded : EMPTY – no restriction*
Permissive ciphers TLS1.3	CIPHERS13	*STRING* - Enumeration separate by colons (OpenSSL format) *Hardcoded : EMPTY – no restriction*
Expected subject of the peer's certificate	PEER_CN	*STRING* - Common name (/CN=xxxxxx) *Hardcoded : EMPTY – peer certificate subject not considered*
Log file name	LOG_FILE	*Specific to K5NETSECURE.DLL* *STRING* - Filemane (with/without path) (if filename without path - located in the Workbench DATA)
Log file level	LOG_LEVEL	*Specific to K5NETSECURE.DLL* *(No log)* *0 (Hardcoded/Default)* *(Information)* 1 *(SSL Message)* 10 *(SSL Protocol)* 11 *(Debug)* 12

3.2. Configuration files – Samples

K5NetSecure.dat

```
[<target_ip_address>:1100]
KEY_CERT=CLIENT.pem
PEER_CN=/CN=SERVER
PUB_CERT=
TLS_MIN=2
TLS_MAX=3
PKI_PATH=C:\Users\Public\Documents\Straton\STRATON 1X.X.X.XXXXX\PKI\CLIENT
CIPHERS=
CIPHERS13=
LOG_LEVEL=10
LOG_FILE= t5tls.log
```

t5secure.dat

```
[*:1100]
KEY_CERT=SERVER.pem
PEER_CN=/CN=CLIENT
PUB_CERT=
TLS_MIN=2
TLS_MAX=3
PKI_PATH=C:\EchangeVM\12.0\PKI\SERVER
CIPHERS=SHA256
CIPHERS13=

[TRACE]
LEVEL=1
```

4. Example of implementation in straton

In the following parts :

- The **CLIENT** will be on **Windows** *(straton workbench)*
- The **SERVER** will be on a **Linux** target *(here a Raspberry)*

Note that if you've followed previous steps, you should have this file architecture :

...\PKI\T5TLS**C**\

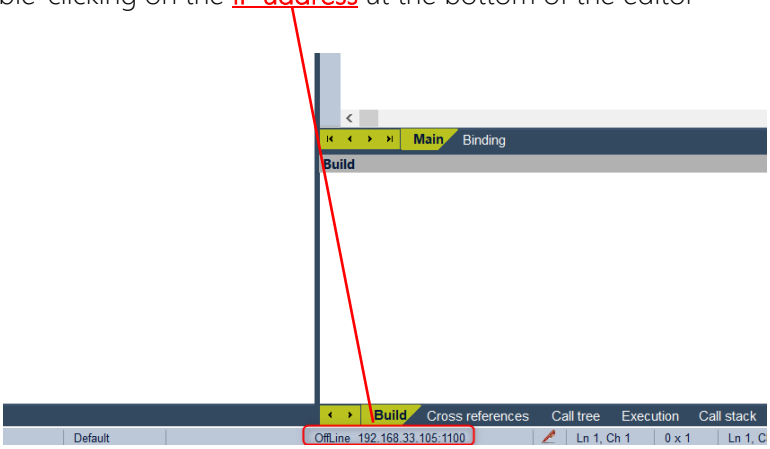
- CA\
  - a23f0115.0 *(trustlist from the CA)*
- PRIVATE\
  - CLIENT.pem *(client's certificate in format PEM + key (*.pem))*

...\PKI\T5TLS**S**\

- CA\
  - a23f0115.0 *(trustlist from the CA)*
- PRIVATE\
  - SERVER.pem *(server's certificate in format PEM + key (*.pem))*

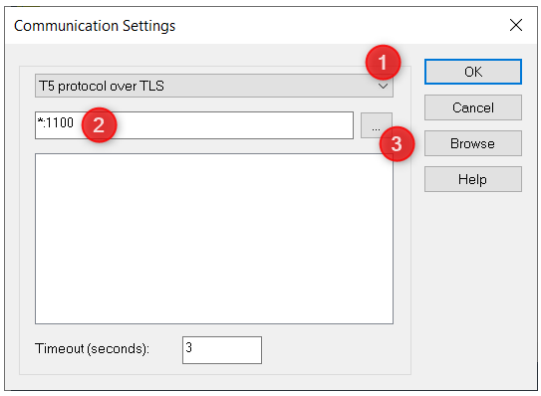
To configure the T5 protocol over TLS go to the "**Communication Settings**".

Can access it by double-clicking on the IP address at the bottom of the editor

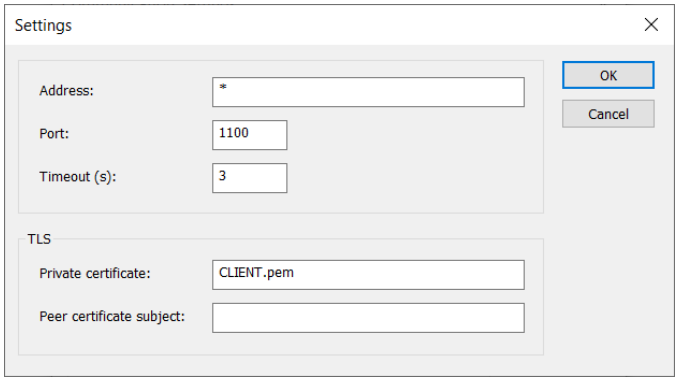


Or from the "**Project**" tab > "**Settings...**" > "**Communication parameters**"

From the following dialog box



1) Select "**T5 protocol over TLS**"
2) Write the IP address of your target following by the communication port number *(here "**:1100**")*
3) Open the advanced parameters *(see below)*



- Put the target certificate name (here "**CLIENT.pem**")
- *(optional)* In the field "**Peer certificate subject**" you can put "**/CN=SERVER**" *(if you've followed the architecture of our example/sample)* to add the certificate Common Name (CN) as an authentication parameter