Raspberry Pi Weather Station Security Setup

For Pro Version of the MyWX Client

V2.4.4

By David W. Enström

Contents

Introduction

This document describes how to create and install certificates to enable a secure connection to your Pi-based weather station over the Internet using the MyWX Pro application, either the Android version or the Windows version. It also describes how to install the certificates to your Pi, your Android device and to a Windows device.

My personal weather station uses the SparkFun weather sensors (Argent Data Systems), and the BC Robotics add-on hardware (HAT) to interface these sensors to the Raspberry Pi.

See: https://www.sparkfun.com/products/8942 and
https://www.bc-robotics.com/tutorials/raspberry-pi-weather-station-part-1/

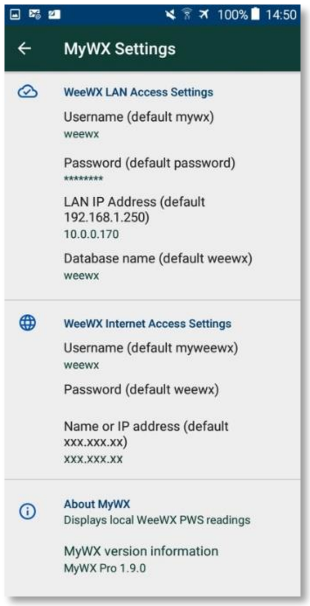
The weather station monitors (and optionally shares the data with Wunderground):

- **Wind Speed** – in km/h
- **Wind Direction** – in degrees
- **Rainfall** – in mm
- **Rain Rate** – in mm/hr
- **Temperature** – in degrees C
- **Air Pressure** – in hecto Pascals
- **Humidity** – in percent

Instructions for constructing this weather station and all the required components are in the "`Weather Station Raspberry Pi construction.pdf`" PDF file. Another document, "`Weather Station Raspberry Pi Software Setup for the MyWX app.pdf`" describes how to install and configure the required software on the Pi, including WeeWX. You can find these documents here:

https://github.com/David-Enst/WeeWX-BCRobotics

MyWX Client



14:50 100%

MyWX Settings

WeeWX LAN Access Settings

Username (default mywx)
weewx

Password (default password)
****

LAN IP Address (default 192.168.1.250)
10.0.0.170

Database name (default weewx)
weewx

WeeWX Internet Access Settings

Username (default myweewx)
weewx

Password (default weewx)

Name or IP address (default xxx.xxx.xx)
xxx.xxx.xx

About MyWX
Displays local WeeWX PWS readings

MyWX version information
MyWX Pro 1.9.0

The MyWX Pro Android / Windows app permits the definition of two types of access and logins to the WeeWX system.

1. A simple username and password access identical to that used in the free version of the app. This login is associated with the local LAN IP address, since it is not very secure; us it only for local LAN access to the device.
2. A very secure PKI (SSL / TLS) based access process. The login is associated with a URL or Internet address. It allows for access to your WeeWX system over the Internet.

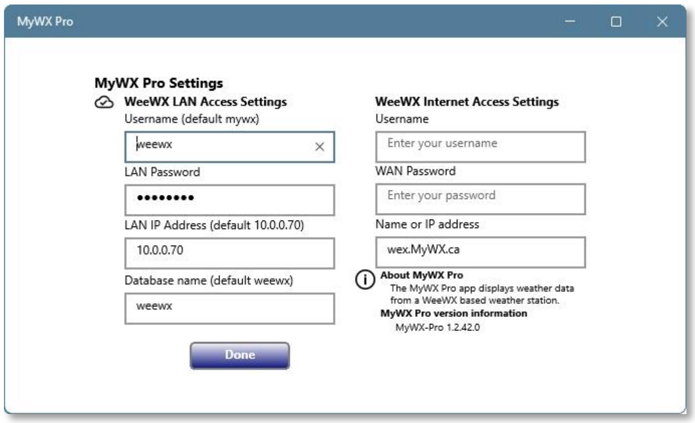
The MyWX Pro Android / Windows app first determines if the LAN address is accessible. If it is, then it attempts to login to the WeeWX database via this LAN-based access method. If this address is not reachable, or this login fails, then the app tries to login via the PKI-based access method.

A view of the settings screen for the MyWX Pro Android app is on the right. Note the two accounts defined to connect to your PWS, one for local LAN connections and one for access via the Internet. Define either or both.

Below is the equivalent Settings screen for the Windows app. Also see:

https://sites.google.com/view/mywx/home

And "MyWX" on the Google store



MyWX Pro Settings

WeeWX LAN Access Settings

Username (default mywx): weewx

LAN Password: ••••••••

LAN IP Address (default 10.0.0.70): 10.0.0.70

Database name (default weewx): weewx

Done

WeeWX Internet Access Settings

Username: Enter your username

WAN Password: Enter your password

Name or IP address: wex.MyWX.ca

About MyWX Pro
The MyWX Pro app displays weather data from a WeeWX based weather station.
MyWX Pro version information
MyWX-Pro 1.2.42.0

Install MySQL (Maria Database)

This section sets up the mandatory MySQL database for MyWX Pro, instead of the default SQLite database. Using the MySQL database is required for multi-user applications (i.e., sharing the data with another client over your local LAN or the Internet).

See the document *Weather Station Raspberry Pi Software Setup for the MyWX App.pdf* here for complete details on this procedure:

https://github.com/David-Enst/WeeWX-BCRobotics

Configuring WeeWX

As noted, the WeeWX configuration *must* specify MySQL instead of SQLite. In the WeeWX configuration file, change the `[[wx_binding]]` section to point to the MySQL database, `archive_mysql`, instead of the SQLite database `archive_sqlite`.

Again, the complete process is described in the document referenced above.

Configuring PKI Security

In order to provide secure remote (from the Internet) access to your device running WeeWX several definitions and configurations need doing.

The process of setting up the required Certificate Authority (CA), server certificates, client certificates, and public keys is rather involved.

A summary of the steps as a preview:

- Define a specific Maria DB username that requires "X.509"
  e.g. GRANT ALL ON weewx.* TO 'myweewx'@'%' REQUIRE X509;
- Define PKI certificates and keys for the Maria DB server;
- Enable SSL/TSL in the Maria DB and pointing to the defined certs and keys;
- Copy the PKI certs to your MyWX Pro client device (e.g., a phone, see below).
- Define port forwarding to your Pi on your Internet router / modem.

Setup the WeeWX Database for Remote Access

Follow these procedures to allow remote access to the MySQL server on your Pi. Note that by default the Maria Data Base only allows local access (i.e., to localhost 127.0.0.1).

In the example below, the local LAN to which the Pi is connected (and your MyWX Android / Windows app and device) has IP addresses in the range `192.168.1.0` to `192.168.1.256`. Adjust as required.

1. In the `/etc/mysql/mariadb.conf.d/50-server.cnf` file change the line as follows to allow remote TCP/IP connections (also make sure that it is not commented out (i.e., `#`):

   ```
   bind-address = *
   ```

2. Create a new MySQL user with required privileges for remote access:

   Start the MariaDB app, and enter these commands (the user account "weewx" is created when weewx is setup and started). Note that the "`%`" character is a wildcard. Adjust the IP address to match your LAN setup.

   ```
   $ sudo mysql -u root –p    // Start the interactive access to the database
   Enter password:            // your root password

   // Definition of the LAN access account (adjust IP address as required):

   MariaDB [(none)]> CREATE USER 'myweewx'@'10.0.0.%' IDENTIFIED BY 'password';
   MariaDB [(none)]> GRANT ALL ON weewx.* TO 'myweewx'@'10.0.0.%';

   // Definition of the WAN (Internet) access account:

   MariaDB [(none)]> CREATE USER 'secweewx'@'%' IDENTIFIED BY 'password';
   MariaDB [(none)]> GRANT ALL ON weewx.* TO 'secweewx'@'%' REQUIRE X509;
   MariaDB [(none)]> FLUSH PRIVILEGES;   // ensures the accesses are updated
   MariaDB [(none)]> exit;
   ```

3. To change a password (if needed):

   ```
   > SET PASSWORD FOR 'weewx'@'10.0.0.%' = PASSWORD('newpass');   or
   > ALTER USER 'weewx'@'localhost' IDENTIFIED BY 'NEWPASSWORD';
   ```

NOTE: There is a difference between `REQUIRE X509` vs. `REQUIRE SSL`:

- ❖ **SSL** – a user can only connect over secured transport (SSL), but still can still authenticate with *username* and *password*
- ❖ **X509** – a user can only connect over secured transport (SSL) *and* the client requires a certificate for authentication.

Define PKI Certificates and Keys on the Pi

Defining certificates and keys for secure access to your Pi is required. One can purchase these from a CA, but for our simple application, we will used self-generated and self-signed certificates, since you control both ends of the remote connection. The one proviso is that we use mutual (2-way) authentication to ensure that both the client (the MyWX Pro app) and the server (the WeeWX/Maria DB) authenticate.

The most convenient way to setup, the required PKI information on your Raspberry Pi is to install a CA Manager. The following assumes that `TinyCA2` is used for this purpose. It simply provides a nice user interface to `OpenSSL`, which typically comes with every Raspberry Pi setup.

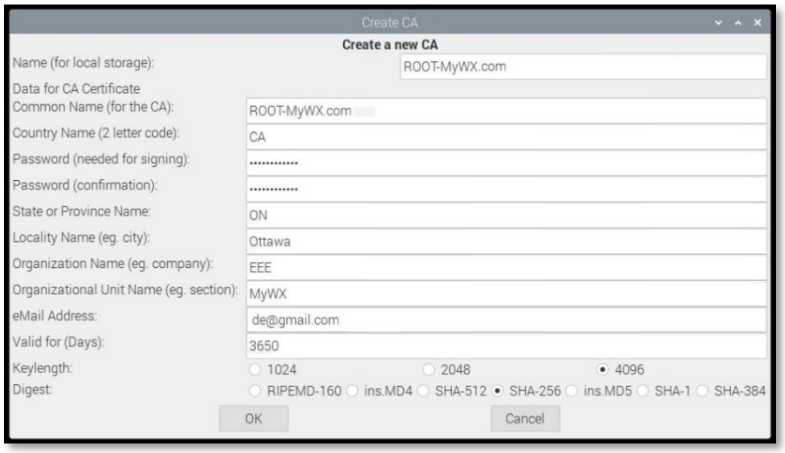
Install `TinyCA2` on your Pi:

```
$ sudo apt-get update -y
$ sudo apt-get install -y tinyca
```

Now run `TinyCA2` on your Pi:

```
$ tinyca2
```

The first step is to define your ROOT CA, from which all other Sub CAs and keys derive. When `TinyCA2` is first run, it opens a window where you define the ROOT CA, as illustrated Figure 1 below.



Create a new CA

Name (for local storage): ROOT-MyWX.com
Data for CA Certificate
Common Name (for the CA): ROOT-MyWX.com
Country Name (2 letter code): CA
Password (needed for signing): ············
Password (confirmation): ············
State or Province Name: ON
Locality Name (eg. city): Ottawa
Organization Name (eg. company): EEE
Organizational Unit Name (eg. section): MyWX
eMail Address: de@gmail.com
Valid for (Days): 3650
Keylength: 1024 2048 [✓ 4096]
Digest: RIPEMD-160 ins.MD4 SHA-512 [✓ SHA-256] ins.MD5 SHA-1 SHA-384

Figure 1 – Create ROOT CA

In this example, note that:

- ➢ The *Name* and *Data for CA Certificate Common Name (for the CA)* should be the same;
- ➢ The Password should be random, long, and difficult to guess. You need this password when creating new Sub CAs.

After clicking `OK`, the CA Configuration window will appear, as shown in Figure 2. This screen defines how the CA will operate. At this point, we make some important decisions.

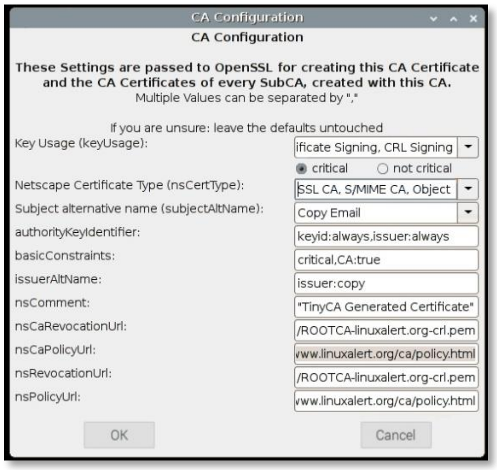
Select weather this CA is `critical` or `not critical`, it is recommended to select `critical`.

Change the Netscape Certificate Type to "`SSL CA, S/MIME CA, Object Signing CA`."

The two revocation URL fields need to be set. This is the address of a web server where the certificates revocation list is accessible to client PCs. To determine if a certificate is valid, this list is consulted to check if a certificate has been revoked. This cannot change after the creation of the CA so you need to have a location setup now. It is recommended that this be hosted on a server separate to the CA itself.

The two policy URL fields relate to a webpage where people can go to read about the policy of the issuing CA, and any certificate use policy you may have. You can make the actual document later, but you must set the web address now.

See the example in Figure 2.



CA Configuration

CA Configuration

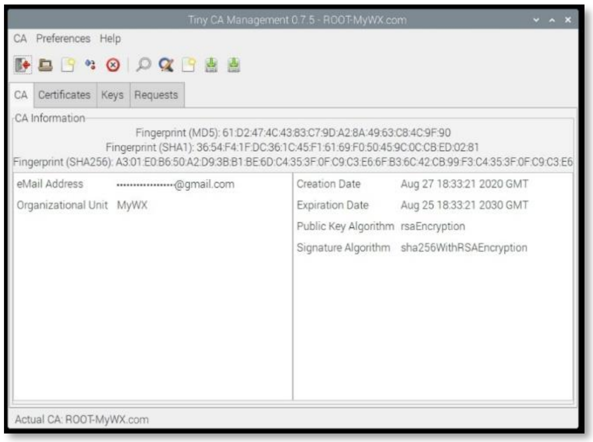
These Settings are passed to OpenSSL for creating this CA Certificate and the CA Certificates of every SubCA, created with this CA.
Multiple Values can be separated by ","

If you are unsure: leave the defaults untouched

Key Usage (keyUsage):	ificate Signing, CRL Signing
	critical [✓] / not critical
Netscape Certificate Type (nsCertType):	SSL CA, S/MIME CA, Object
Subject alternative name (subjectAltName):	Copy Email
authorityKeyIdentifier:	keyid:always,issuer:always
basicConstraints:	critical,CA:true
issuerAltName:	issuer:copy
nsComment:	"TinyCA Generated Certificate"
nsCaRevocationUrl:	/ROOTCA-linuxalert.org-crl.pem
nsCaPolicyUrl:	vww.linuxalert.org/ca/policy.html
nsRevocationUrl:	/ROOTCA-linuxalert.org-crl.pem
nsPolicyUrl:	vww.linuxalert.org/ca/policy.html

Figure 2 – CA Configuration

Click `OK` to accept the settings. After creating the CA, which may take some time, the main `TinyCA` window, showing your root CA, opens as shown in Figure 3.



Tiny CA Management 0.7.5 - ROOT-MyWX.com

CA Information

Fingerprint (MD5): 61:D2:47:4C:43:83:C7:9D:A2:8A:49:63:C8:4C:9F:90
Fingerprint (SHA1): 36:54:F4:1F:DC:36:1C:45:F1:61:69:F0:50:45:9C:0C:CB:ED:02:81
Fingerprint (SHA256): A3:01:E0:B6:50:A2:D9:3B:B1:BE:6D:C4:35:3F:0F:C9:C3:E6:6F:B3:6C:42:CB:99:F3:C4:35:3F:0F:C9:C3:E6

eMail Address	..................@gmail.com	Creation Date	Aug 27 18:33:21 2020 GMT
Organizational Unit	MyWX	Expiration Date	Aug 25 18:33:21 2030 GMT
		Public Key Algorithm	rsaEncryption
		Signature Algorithm	sha256WithRSAEncryption

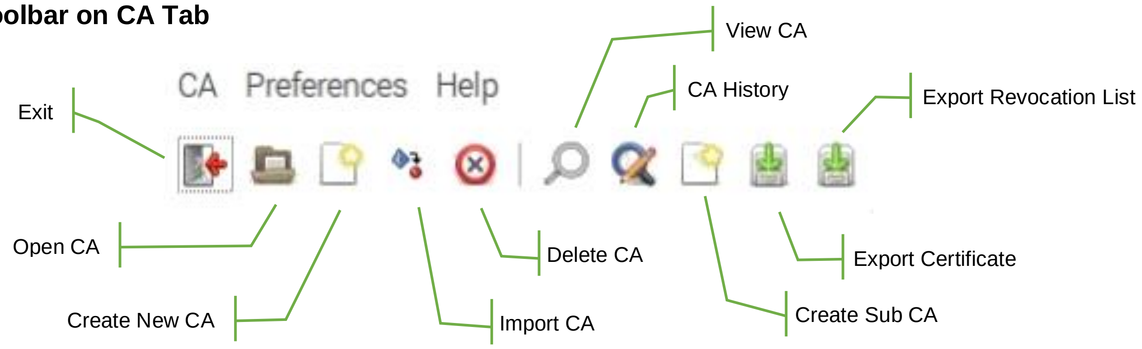
Actual CA: ROOT-MyWX.com

Figure 3 – CA Management Showing Root Certificate

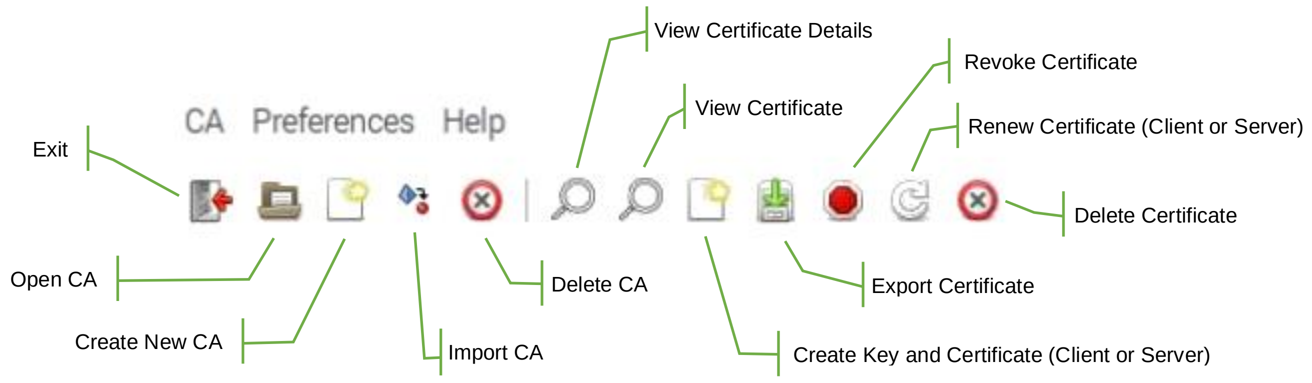
Now that we have the ROOT CA defined, we can create the Subordinate CA, which is what we use within the Pi / Maria DB, and the MyWX Pro app on your device. You may use this root CA, however it is highly recommended to define a Sub CA (more flexible, among other reasons).

One annoying thing you will notice with `TinyCA2` is that none of the buttons on the toolbar has captions or tooltips. Very annoying, however it is very safe to click each one to see what it does since you can cancel every screen to get back. A summary of these Icons and what they do for each tab:

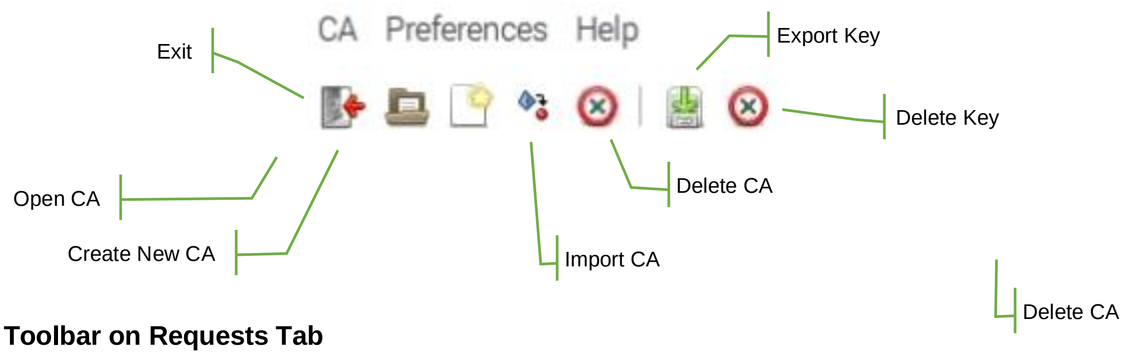
TinyCA – Toolbar on CA Tab



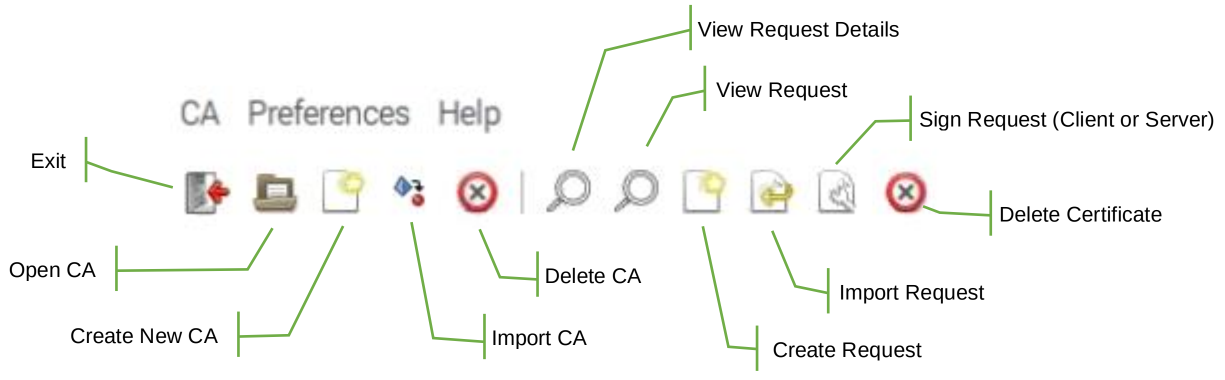
TinyCA – Toolbar on Certificates Tab



TinyCA – Toolbar on Keys Tab



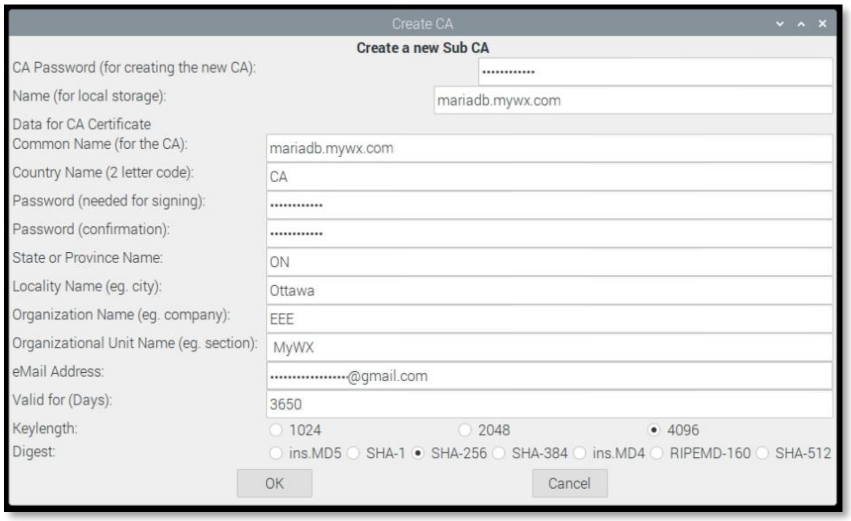
TinyCA – Toolbar on Requests Tab



To create the Sub CA:

1. Make sure you have TinyCA2 open with your Root CA displayed (see Figure 3). On the action icons, click the “Create a new Sub CA” button, which is the 3rd button from the right.
2. Fill in the details for the Sub CA (see Figure 4). The CA Password field is the password of the root CA (Or in the case of multiple Sub CA levels, the password of the parent CA). The second password asked for is a password for this Sub CA, which needs to be different from the root-CA password. This password is used for signing client certificates.
3. Click OK when satisfied. The Sub CA is created and you are returned to the main application window with the Sub CA opened (see Figure 5).

Here is an example definition:



Create CA

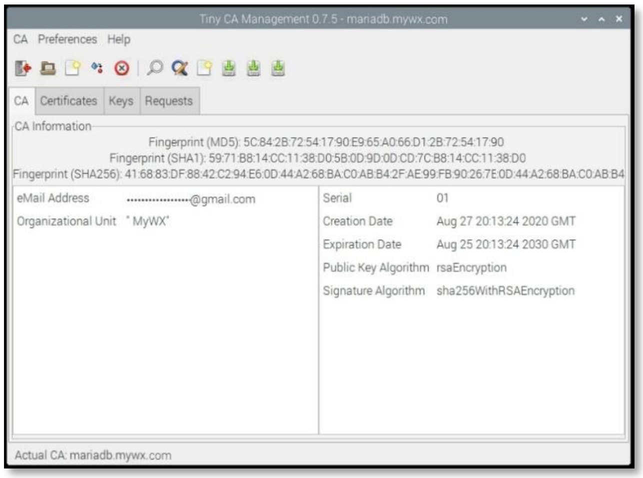
Create a new Sub CA

CA Password (for creating the new CA):	············
Name (for local storage):	mariadb.mywx.com
Data for CA Certificate	
Common Name (for the CA):	mariadb.mywx.com
Country Name (2 letter code):	CA
Password (needed for signing):	············
Password (confirmation):	············
State or Province Name:	ON
Locality Name (eg. city):	Ottawa
Organization Name (eg. company):	EEE
Organizational Unit Name (eg. section):	MyWX
eMail Address:	··················@gmail.com
Valid for (Days):	3650
Keylength:	1024 2048 [✓ 4096]
Digest:	ins.MD5 SHA-1 [✓ SHA-256] SHA-384 ins.MD4 RIPEMD-160 SHA-512

Figure 4 – Create a Sub CA

***Important**: Make sure that you keep track of all these passwords; they are needed in order to export and make use of these Certificates and Keys!*

An example main application window with the Sub CA opened:



Tiny CA Management 0.7.5 - mariadb.mywx.com

CA Information

Fingerprint (MD5): 5C:84:2B:72:54:17:90:E9:65:A0:66:D1:2B:72:54:17:90
Fingerprint (SHA1): 59:71:B8:14:CC:11:38:D0:5B:0D:9D:0D:CD:7C:B8:14:CC:11:38:D0
Fingerprint (SHA256): 41:68:83:DF:88:42:C2:94:E6:0D:44:A2:68:BA:C0:AB:B4:2F:AE:99:FB:90:26:7E:0D:44:A2:68:BA:C0:AB:B4

eMail Address	··················@gmail.com	Serial	01
Organizational Unit	" MyWX"	Creation Date	Aug 27 20:13:24 2020 GMT
		Expiration Date	Aug 25 20:13:24 2030 GMT
		Public Key Algorithm	rsaEncryption
		Signature Algorithm	sha256WithRSAEncryption

Actual CA: mariadb.mywx.com

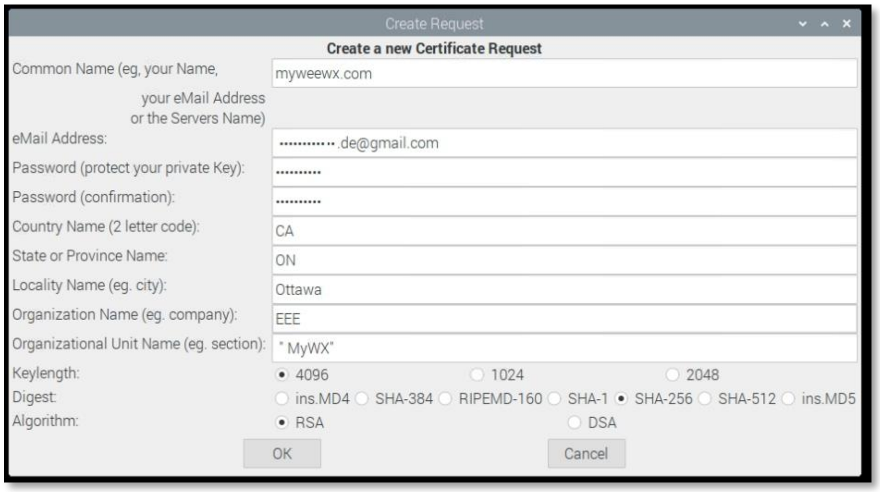
Figure 5 – Sub CA Window

Now, we use the Sub CA definition to create certificates and keys and then export them for use with the Maria DB and the MyWX Pro Android / Windows app. For this new CA hierarchy to be of any use, the required trust by the devices that use its services is defined.

Create Server Certificate and Key

Now its time to create the certificate for the Maria database ***server***. Ensure that you have the Sub CA (e.g., `mariadb.mywx.ca`) open, as shown in Figure 5. Select the `Requests` tab. Right click and select "`New`

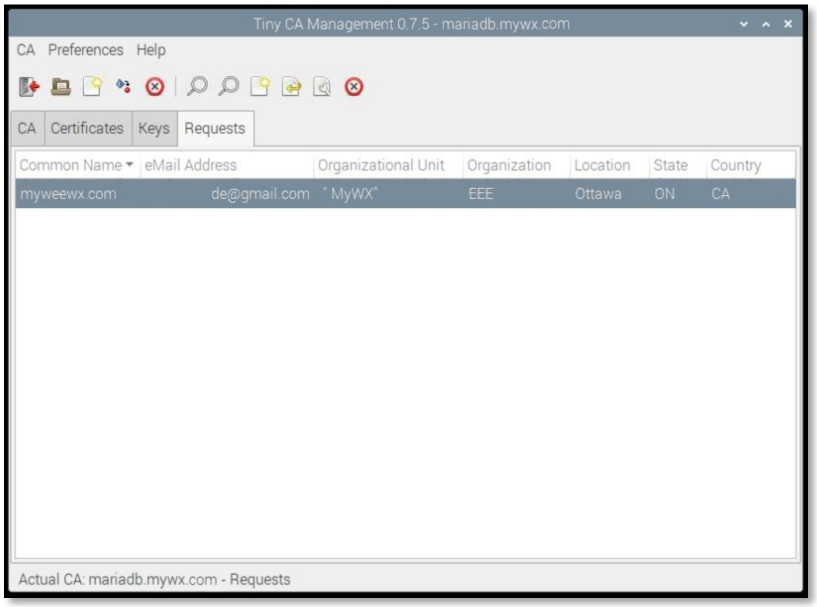
`Request`", or click the New Request icon. You get a new window called `Create Request` as shown in Figure 6. The common name must be the same name as the one used to access your server, in my case `myweex.ca`. Once again, you have to enter a password and once again, it must *not* be the same as the previous passwords. Fill in the form, similar to the example below, and press OK.


Create Request

Create a new Certificate Request

Common Name (eg, your Name, your eMail Address or the Servers Name)	myweewx.com
eMail Address:	············.de@gmail.com
Password (protect your private Key):	··········
Password (confirmation):	··········
Country Name (2 letter code):	CA
State or Province Name:	ON
Locality Name (eg. city):	Ottawa
Organization Name (eg. company):	EEE
Organizational Unit Name (eg. section):	" MyWX"
Keylength:	[✓ 4096] 1024 2048
Digest:	ins.MD4 SHA-384 RIPEMD-160 SHA-1 [✓ SHA-256] SHA-512 ins.MD5
Algorithm:	[✓ RSA] DSA

Figure 6 – Create Sub CA Server Certificate Request

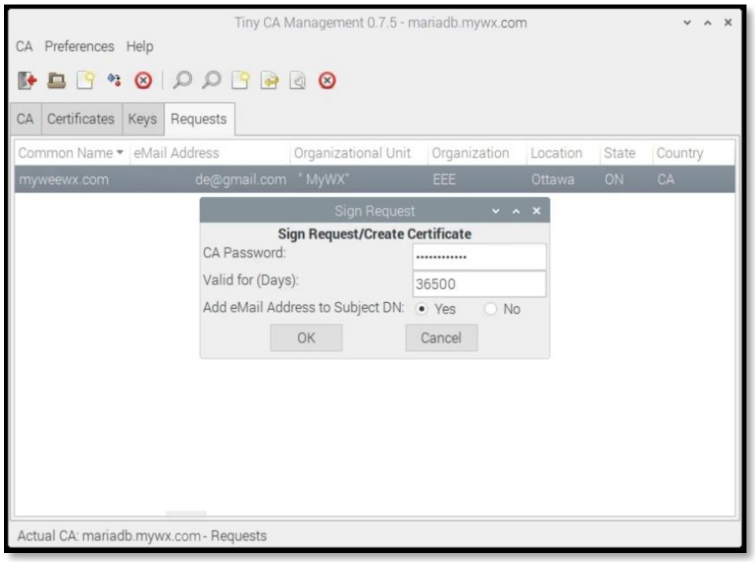

Tiny CA Management 0.7.5 - mariadb.mywx.com

Common Name	eMail Address	Organizational Unit	Organization	Location	State	Country
myweewx.com	de@gmail.com	" MyWX"	EEE	Ottawa	ON	CA

Actual CA: mariadb.mywx.com - Requests

Figure 7 – Sub CA Request Tab

The new certificate Request is listed in the `Requests` tab, as shown in Figure 7. Right click on the Request and select "`Sign Request`". You are prompted to define if it is a *server* or *client* request. In this case, select ***server***. You get a new window asking for the CA password (i.e., the Sub CA password). It is now that you, as a CA, approve the request and confirm that the certificate holder is the correct owner of the common name in the certificate and the rest of the information in the certificate is correct. At the same time, you decide for how long time the certificate is valid. Enter your Sub CA's password and press OK. You have now signed your certificate, which creates the required server Certificate and Key.


Tiny CA Management 0.7.5 - mariadb.mywx.com

Common Name	eMail Address	Organizational Unit	Organization	Location	State	Country
myweewx.com	de@gmail.com	"MyWX"	EEE	Ottawa	ON	CA

Sign Request/Create Certificate

CA Password: ....

Valid for (Days): 36500

Add eMail Address to Subject DN: [✓ Yes] No

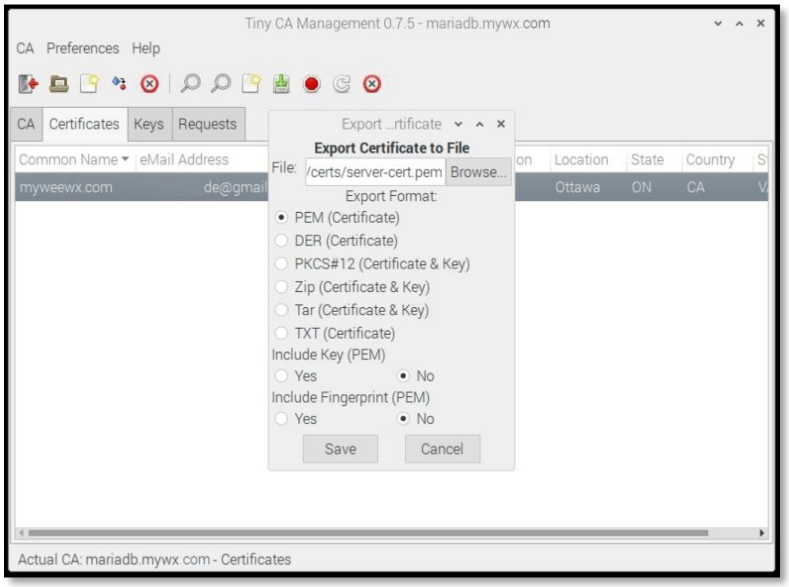
Actual CA: mariadb.mywx.com - Requests

Figure 8 – Sub CA, Sign Server Request

Now we will:

- ➢ Export the Certificate and Key for the Maria DB server (two **.PEM** files)
- ➢ Export the Certificate and Key for use with the MyWX Pro app (a single **.P12** file)
- ➢ Export the Sub CA certificate chain (a **.PEM** file)

Return to `TinyCA2` and verify that you have your Sub CA opened. Select the `Certificates` tab, right click on the server’s certificate and select `Export Certificate`. You may change the file name, then click `Save`. I use the filename `server-cert.pem`. The Maria DB uses this **PEM** file.


Tiny CA Management 0.7.5 - mariadb.mywx.com

Export Certificate to File

File: /certs/server-cert.pem

Export Format: [✓ PEM (Certificate)] DER (Certificate) PKCS#12 (Certificate & Key) Zip (Certificate & Key) Tar (Certificate & Key) TXT (Certificate)

Include Key (PEM): Yes [✓ No]

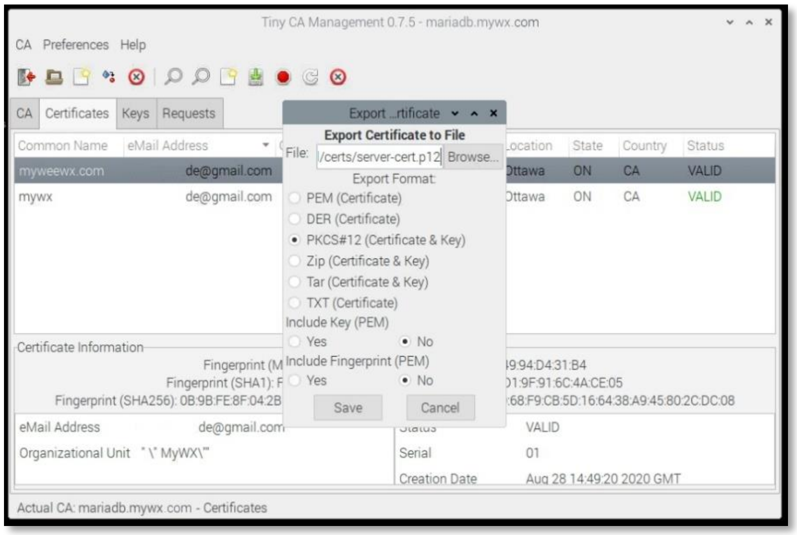
Include Fingerprint (PEM): Yes [✓ No]

Actual CA: mariadb.mywx.com - Certificates

Figure 9 – Export Server Certificate (PEM)

Now we will export the ***server*** Certificate and Key for use with the MyWX Pro app. Select the Certificates tab, right click on the cert and select `Export Certificate`. In this case, the ***file name must be*** `server-cert.p12`.

Note that a **`P12`** format file includes both the Certificate information *and* the Key. The Weewx Pi app uses this **`P12`** file. See Figure 10.


Tiny CA Management 0.7.5 - mariadb.mywx.com
CA Preferences Help
CA Certificates Keys Requests
Export ..rtificate
Save
Cancel
Browse...

Export Certificate to File

File: /certs/server-cert.p12

Export Format:
- PEM (Certificate)
- DER (Certificate)
- [✓ PKCS#12 (Certificate & Key)]
- Zip (Certificate & Key)
- Tar (Certificate & Key)
- TXT (Certificate)

Include Key (PEM): Yes [✓ No]

Include Fingerprint (PEM): Yes [✓ No]

Common Name	eMail Address	Location	State	Country	Status
myweewx.com	de@gmail.com	Ottawa	ON	CA	VALID
mywx	de@gmail.com	Ottawa	ON	CA	VALID

Certificate Information

Fingerprint (M… 9:94:D4:31:B4

Fingerprint (SHA1): F… 01:9F:91:6C:4A:CE:05

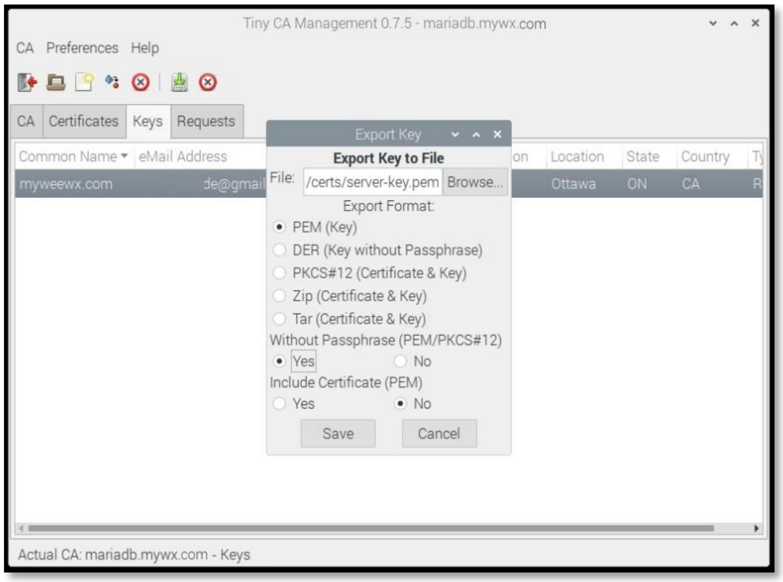
Fingerprint (SHA256): 0B:9B:FE:8F:04:2B… 68:F9:CB:5D:16:64:38:A9:45:80:2C:DC:08

eMail Address	de@gmail.com	Status	VALID
Organizational Unit	"\" MyWX\""	Serial	01
		Creation Date	Aug 28 14:49:20 2020 GMT

Actual CA: mariadb.mywx.com - Certificates

Figure 10 – Export Cert and Key for MyWX Pro (P12)

Now do the same with the Key. Select the `Keys` tab, right click on the key and select `Export Key`. You may change the file name. I use the name `server-cert.pem` as filename. Also, set "`Without Passphrase`" to `Yes`. This lets you start the web server without enter the certificates password. This lets you start the web server without enter the certificates password. Then click `Save`. TinyCA2 will ask you for the password of the certificate for the web server certificate / key. The Maria DB uses this **`PEM`** file.


Tiny CA Management 0.7.5 - mariadb.mywx.com
CA Preferences Help
CA Certificates Keys Requests
Export Key
Browse...
Save
Cancel

Export Key to File

File: /certs/server-key.pem

Export Format:
- [✓ PEM (Key)]
- DER (Key without Passphrase)
- PKCS#12 (Certificate & Key)
- Zip (Certificate & Key)
- Tar (Certificate & Key)

Without Passphrase (PEM/PKCS#12): [✓ Yes] No

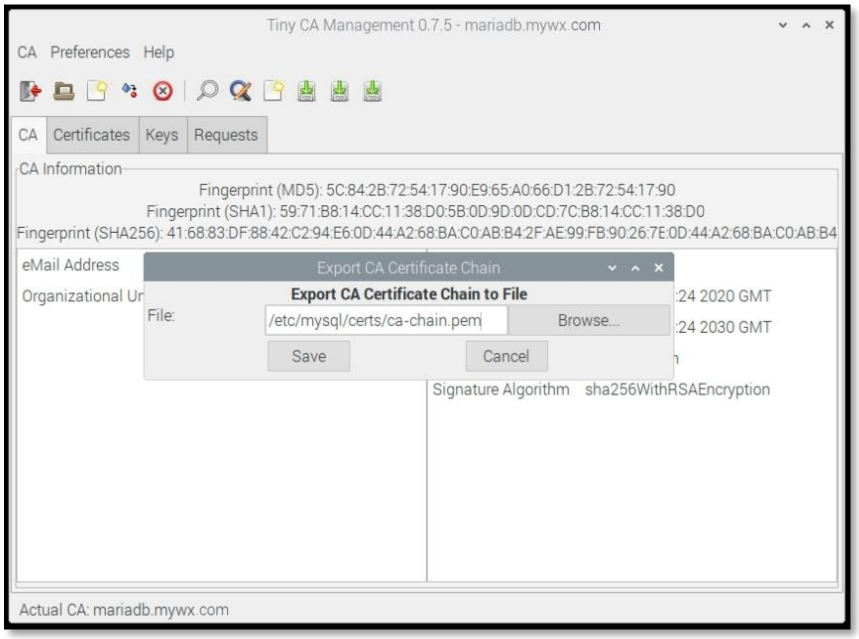
Include Certificate (PEM): Yes [✓ No]

Common Name	eMail Address	Location	State	Country
myweewx.com	de@gmail…	Ottawa	ON	CA

Actual CA: mariadb.mywx.com - Keys

Figure 11 – Export Server Key (PEM)

We now export the certificate chain for use with the Maria DB. Go to the CA tab and select the right-most icon in the tool bar and save the file. Select a location and name (e.g., `/etc/mysql/certs/ca-chain.pem`), and click `Save`. The Maria DB uses this **`PEM`** file.


Tiny CA Management 0.7.5 - mariadb.mywx.com
CA Preferences Help
CA Certificates Keys Requests
CA Information

Fingerprint (MD5): 5C:84:2B:72:54:17:90:E9:65:A0:66:D1:2B:72:54:17:90
Fingerprint (SHA1): 59:71:B8:14:CC:11:38:D0:5B:0D:9D:0D:CD:7C:B8:14:CC:11:38:D0
Fingerprint (SHA256): 41:68:83:DF:88:42:C2:94:E6:0D:44:A2:68:BA:C0:AB:B4:2F:AE:99:FB:90:26:7E:0D:44:A2:68:BA:C0:AB:B4

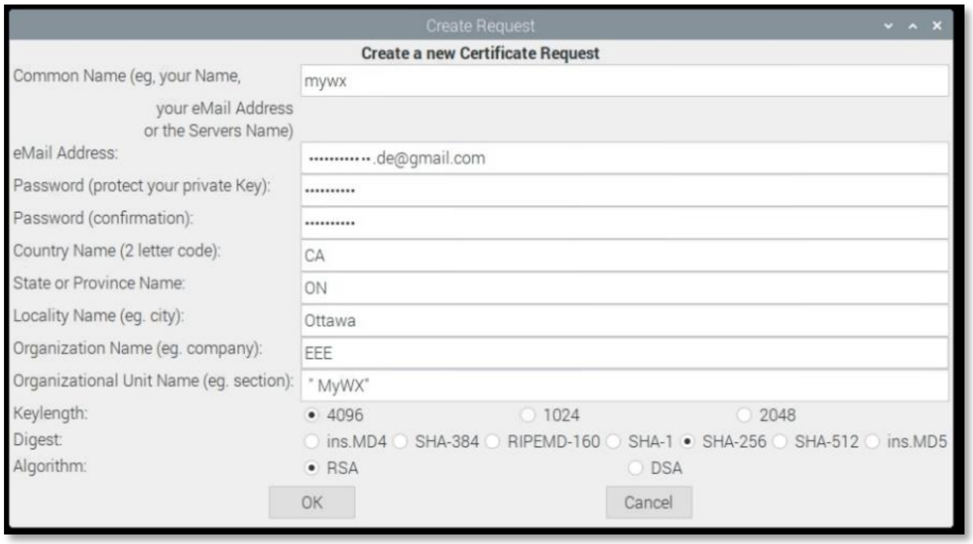
Export CA Certificate Chain
Export CA Certificate Chain to File
File: /etc/mysql/certs/ca-chain.pem

Signature Algorithm sha256WithRSAEncryption

Figure 12 – Export Sub CA Certificate Chain (PEM)

Create Client Certificate and Key

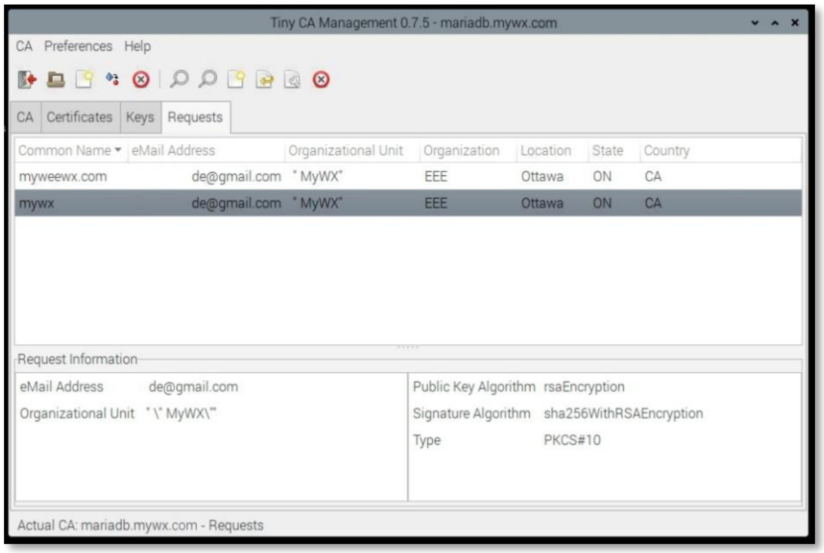
Now we create the certificate for the Maria database ***clients***. Ensure that you have the Sub CA (e.g., `mariadb.mywx.ca`) open, as shown in Figure 5. Select the `Requests` tab. Right click and select "`New Request`", or click the New Request icon. You get a new window called `Create Request`. The common name should be the same name as the one used to access your server, in my case `MyWX`. Once again, you have to enter a password and once again, it must *not* be the same as the previous passwords. Fill in the form, similar to the example below, and press OK.



Create a new Certificate Request

Common Name (eg, your Name, your eMail Address or the Servers Name)	mywx
eMail Address:	••••••••••••.de@gmail.com
Password (protect your private Key):	••••••••••
Password (confirmation):	••••••••••
Country Name (2 letter code):	CA
State or Province Name:	ON
Locality Name (eg. city):	Ottawa
Organization Name (eg. company):	EEE
Organizational Unit Name (eg. section):	" MyWX"
Keylength:	● 4096 ○ 1024 ○ 2048
Digest:	○ ins.MD4 ○ SHA-384 ○ RIPEMD-160 ○ SHA-1 ● SHA-256 ○ SHA-512 ○ ins.MD5
Algorithm:	● RSA ○ DSA

Figure 13 – Create Client Certificate Request


Tiny CA Management 0.7.5 - mariadb.mywx.com
CA Preferences Help
CA Certificates Keys Requests

Common Name	eMail Address	Organizational Unit	Organization	Location	State	Country
myweewx.com	de@gmail.com	" MyWX"	EEE	Ottawa	ON	CA
mywx	de@gmail.com	" MyWX"	EEE	Ottawa	ON	CA

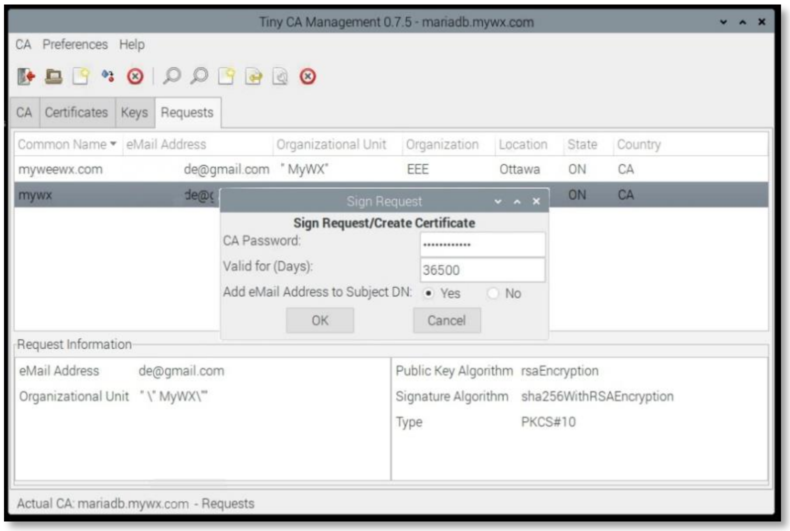
Request Information

eMail Address	de@gmail.com
Organizational Unit	" \" MyWX\""
Public Key Algorithm	rsaEncryption
Signature Algorithm	sha256WithRSAEncryption
Type	PKCS#10

Actual CA: mariadb.mywx.com - Requests

Figure 14 –Client (and Server) Certificate Requests

The new certificate request is listed in the `Requests` tab, as shown in Figure 14. Right click on the Request and select "`Sign Request`". You are prompted to define if it is a *server* or *client* request. In this case, select ***client***. You get a new window asking for the CA password (i.e., the Sub CA password). It is now that you, as a CA, approve the request and confirm that the certificate holder is the correct owner of the common name in the certificate and the rest of the information in the certificate is correct. At the same time, you decide for how long the certificate is valid. Enter your Sub CA's password and press OK. You have now signed your certificate and created the required client certificate and key.


Tiny CA Management 0.7.5 - mariadb.mywx.com
CA Preferences Help
CA Certificates Keys Requests
Sign Request
OK
Cancel

Sign Request/Create Certificate

CA Password: ············

Valid for (Days): 36500

Add eMail Address to Subject DN: [✓ Yes] No

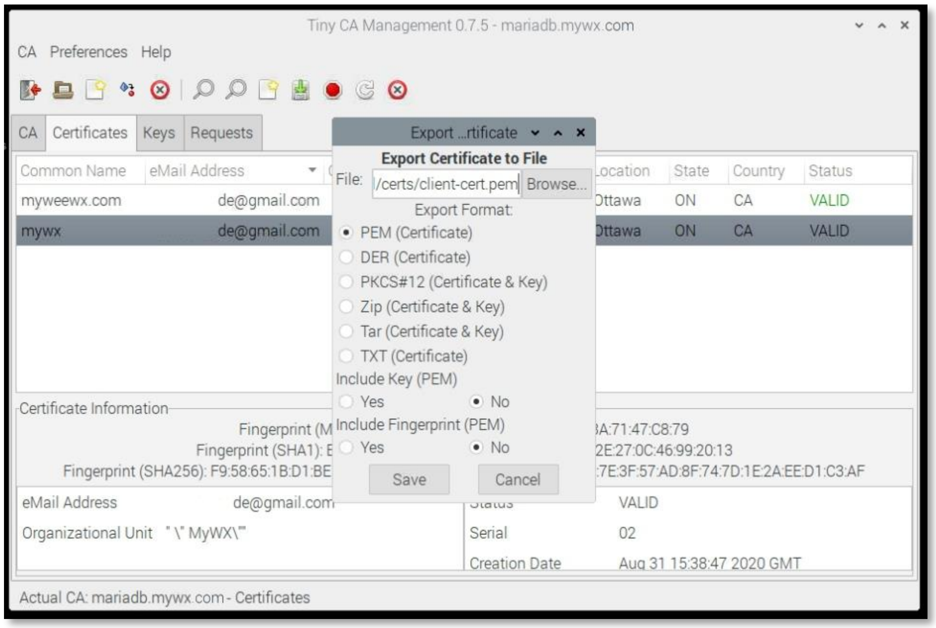
Actual CA: mariadb.mywx.com - Requests

Figure 15 – Sign Client Certificate Request

Now we will:

- ➢ Export the Certificate and Key for the Maria DB clients (two **`.PEM`** files)
- ➢ Export the Certificate and Key for use with the MyWX Pro Android / Windows app (a single **`.P12`** file)

Return to TinyCA2 and verify that you have your Sub CA opened. Select the `Certificates` tab, right click on the client's certificate and select `Export Certificate`. You may change the file name, then click `Save`, as shown in Figure 16. I use the filename `client-cert.pem`. The Maria DB uses this **`PEM`** file.



Tiny CA Management 0.7.5 - mariadb.mywx.com

CA Preferences Help

CA | Certificates | Keys | Requests

Common Name	eMail Address	Location	State	Country	Status
myweewx.com	de@gmail.com	Ottawa	ON	CA	VALID
mywx	de@gmail.com	Ottawa	ON	CA	VALID

Export ...rtificate

Export Certificate to File

File: /certs/client-cert.pem Browse...

Export Format:
- [✓ PEM (Certificate)]
- DER (Certificate)
- PKCS#12 (Certificate & Key)
- Zip (Certificate & Key)
- Tar (Certificate & Key)
- TXT (Certificate)

Include Key (PEM): Yes [✓ No]

Include Fingerprint (PEM): Yes [✓ No]

Save Cancel

Certificate Information

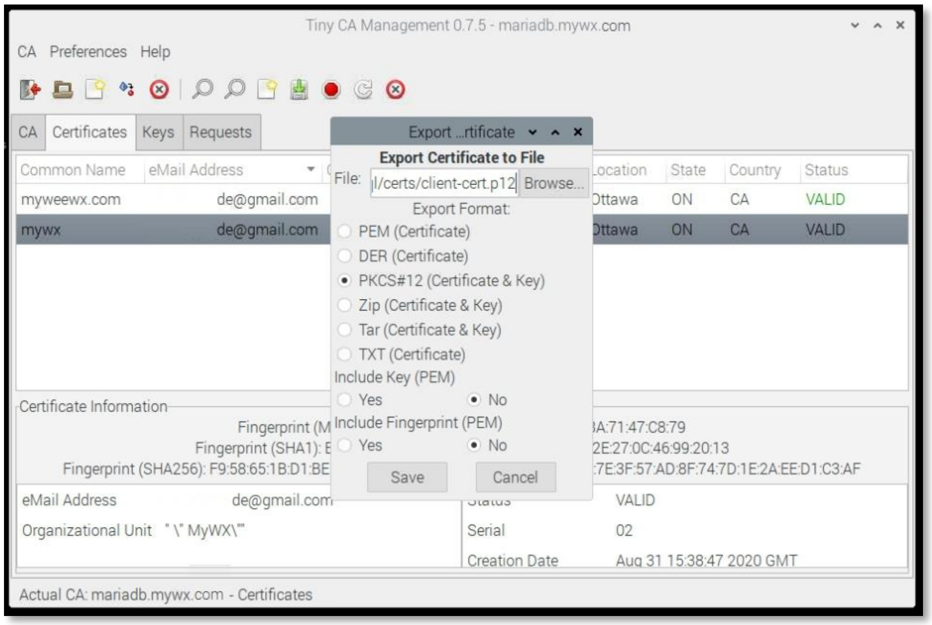
Fingerprint (M ... 3A:71:47:C8:79
Fingerprint (SHA1): E ... 2E:27:0C:46:99:20:13
Fingerprint (SHA256): F9:58:65:1B:D1:BE ... 7E:3F:57:AD:8F:74:7D:1E:2A:EE:D1:C3:AF

eMail Address	de@gmail.com	Status	VALID
Organizational Unit	" \" MyWX\""	Serial	02
		Creation Date	Aug 31 15:38:47 2020 GMT

Actual CA: mariadb.mywx.com - Certificates

Figure 16 – Export Client Certificate (PEM)

Now we will export the ***client*** Certificate and Key for use with the MyWX Pro Android / Windows app. Select the Key tab, right click on the key and select `Export Key`. In this case, the file name *must* be **`client-cert.p12`**. Also, the "Friendly Name" for the certificate *must* be "***MyWX-Client***" since this is what the Windows app uses to find the certificate. Note that a **`P12`** format file includes both the Certificate information *and* the Key; see Figure 17.



Tiny CA Management 0.7.5 - mariadb.mywx.com

CA Preferences Help

CA | Certificates | Keys | Requests

Common Name	eMail Address	Location	State	Country	Status
myweewx.com	de@gmail.com	Ottawa	ON	CA	VALID
mywx	de@gmail.com	Ottawa	ON	CA	VALID

Export ...rtificate

Export Certificate to File

File: l/certs/client-cert.p12 Browse...

Export Format:
- PEM (Certificate)
- DER (Certificate)
- [✓ PKCS#12 (Certificate & Key)]
- Zip (Certificate & Key)
- Tar (Certificate & Key)
- TXT (Certificate)

Include Key (PEM): Yes [✓ No]

Include Fingerprint (PEM): Yes [✓ No]

Save Cancel

Certificate Information

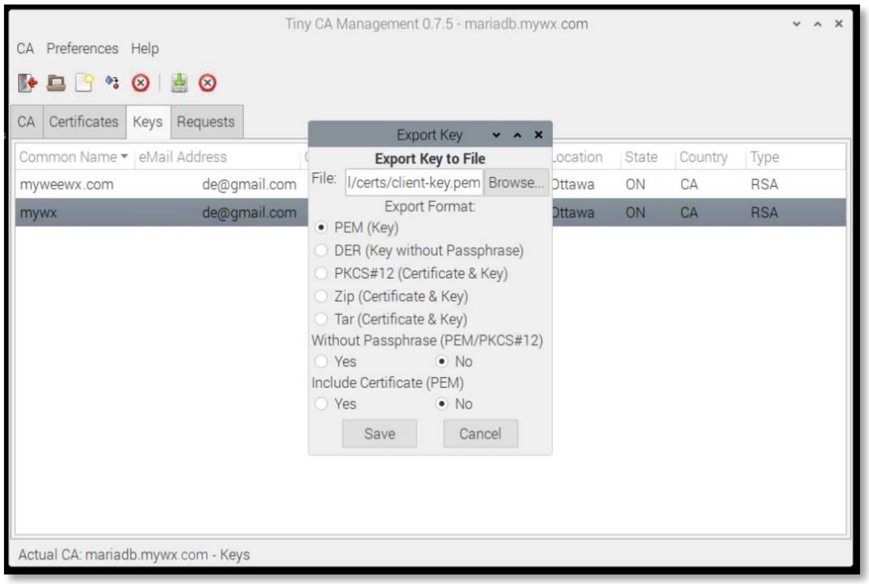
Fingerprint (M ... 3A:71:47:C8:79
Fingerprint (SHA1): E ... 2E:27:0C:46:99:20:13
Fingerprint (SHA256): F9:58:65:1B:D1:BE ... 7E:3F:57:AD:8F:74:7D:1E:2A:EE:D1:C3:AF

eMail Address	de@gmail.com	Status	VALID
Organizational Unit	" \" MyWX\""	Serial	02
		Creation Date	Aug 31 15:38:47 2020 GMT

Actual CA: mariadb.mywx.com - Certificates

Figure 17 – Export Client Certificate and Key (P12)

Now do the same with the Key. Select the `Keys` tab, right click on the key and select `Export Key`. You may change the file name. I use the name `client-key.pem` as the filename. Also, set “`Without Passphrase`” to `Yes`. This lets you start the web server without enter the certificate password. Then click `Save`. TinyCA2 will ask you for the password of the certificate for the web client certificate / key. The Maria DB uses this **`PEM`** file, see Figure 17Figure 18.



Tiny CA Management 0.7.5 - mariadb.mywx.com
CA Preferences Help
CA Certificates Keys Requests
Actual CA: mariadb.mywx.com - Keys

Common Name	eMail Address	Location	State	Country	Type
myweewx.com	de@gmail.com	Ottawa	ON	CA	RSA
mywx	de@gmail.com	Ottawa	ON	CA	RSA

Export Key

Export Key to File

File: /certs/client-key.pem Browse...

Export Format:
[✓ PEM (Key)]
DER (Key without Passphrase)
PKCS#12 (Certificate & Key)
Zip (Certificate & Key)
Tar (Certificate & Key)

Without Passphrase (PEM/PKCS#12)
Yes [✓ No]

Include Certificate (PEM)
Yes [✓ No]

Save Cancel

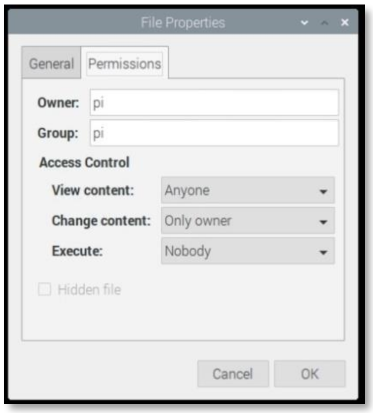
Figure 18 – Export Client Key (PEM)

Point the Maria DB to the Certificates and Keys

The required files were saved to the certs folder on your Raspberry Pi, as described.

The other file created, the `server-cert.p12` file, we will copy to your Android device for use in authenticating the Server side.

***Note**:* *You **must** ensure that the permissions on these generated files are properly defined on your Raspberry Pi; see below.*



Owner: pi
Group: pi

Access Control
View content: Anyone
Change content: Only owner
Execute: Nobody

Hidden file

Server Definitions

The three files used by the Maria DB server, to provide TLS security for the remote access to the WeeWX database, are:

/etc/mysql/certs/ca-chain.pem	– defines the base trust
/etc/mysql/certs/server-cert.pem	– defines the certificate for the server (public key)
/etc/mysql/certs/server-key.pem	– defines the key for the server (private key)

These three server certificate and key files, used by the Maria DB server, are referenced in a configuration file, typically:

```
/etc/mysql/mariadb.conf.d/50-server.cnf
```

Modify this section of file as shown (uncomment lines and adjust filenames):

```
# * Security Features
#
# Read the manual, too, if you want chroot!
#chroot = /var/lib/mysql/
#
# Use the GUI tool "tinyca" for creating TLS certs and keys.
#
ssl-ca = /etc/mysql/ca-chain.pem
ssl-cert = /etc/mysql/server-cert.pem
ssl-key = /etc/mysql/server-key.pem
#
# Accept only connections using secure TLS protocols
#   ...when MariaDB is compiled with OpenSSL:
#
#ssl-cipher = TLSv1.2
#
#   ...when MariaDB is compiled with YaSSL (default in Debian):
#
ssl = on
```

Note that the last line requires TLS on all remote accesses!

Client Definitions

The two files, used by the Maria DB clients, are referenced in a configuration file, typically:

```
/etc/mysql/mariadb.conf.d/50-client.cnf
```

Modify this section of file as shown (uncomment lines and adjust filenames):

```
# Example of client certificate usage
ssl-cert = /etc/mysql/certs/client-cert.pem
ssl-key  = /etc/mysql/certs/client-key.pem
#
# Allow only TLS encrypted connections
#ssl-verify-server-cert=on
```

Secure the Maria DB Installation

One final step is to define a secure installation for the Maria DB. Execute the following script and answer the prompts. It enables you to improve the security of your Maria DB installation in the following ways:

- You can set a password for root accounts;
- You can remove root accounts that are accessible from outside the local host;
- You can remove anonymous-user accounts;
- You can remove the test database, which by default can be accessed by anonymous users.

Important: Run this script in order to avoid issues with your ISP!

```
dave@Pi:~ $ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
 ... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
 ... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
```

```
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
dave@Pi:~ $
```

Now you can enable security by re-starting MySQL:

```
$ sudo service mysql restart
```

Now check the setup:

```
$ sudo mysql -u root -p
MariaDB [(none)]> SHOW VARIABLES LIKE '%ssl%';

+---------------------+----------------------------------+
| Variable_name       | Value                            |
+---------------------+----------------------------------+
| have_openssl        | NO                               |
| have_ssl            | YES                              |
| ssl_ca              | /etc/mysql/certs/ca-chain.pem    |
| ssl_capath          |                                  |
| ssl_cert            | /etc/mysql/certs/server-cert.pem |
| ssl_cipher          |                                  |
| ssl_crl             |                                  |
| ssl_crlpath         |                                  |
| ssl_key             | /etc/mysql/certs/server-key.pem  |
| version_ssl_library | YaSSL 2.4.4                      |
+---------------------+----------------------------------+

Or perhaps this result:
+---------------------+----------------------------------+
| Variable_name       | Value                            |
+---------------------+----------------------------------+
| have_openssl        | Yes                               |
| have_ssl            | YES                              |
| ssl_ca              | /etc/mysql/certs/ca-chain.pem    |
| ssl_capath          |                                  |
| ssl_cert            | /etc/mysql/certs/server-cert.pem |
| ssl_cipher          | TLSv1.2                          |
| ssl_crl             |                                  |
| ssl_crlpath         |                                  |
| ssl_key             | /etc/mysql/certs/server-key.pem  |
| version_ssl_library | OpenSSL 1.1.1n  15 Mar 2022      |
+---------------------+----------------------------------+
```

> **Note:** `If the Maridb was compiled with OpenSSL then you need to update the 50-server.cnf file (i.e., uncomment the line #ssl-cipher = TLSv1.2)`

As you can see, `have_ssl` says `YES`, therefore PKI security is now enabled.

Installing CA Certificate on your Android Phone

The MyWX Pro app and Maria DB (WeeWX) app are setup to authenticate each other, ensuring that a man-in-the-middle attack is thwarted. To do this you must copy the CA certificates (i.e., the `client-cert.p12` file and the `server-cert.p12` file) to your phone (or other Android devices) on which the MyWX Pro app is installed.

- ➢ Copy the `client-cert.p12` file and the `server-cert.p12` to your Android device. They ***must*** be copied to the `Android\data\com.unified.mywx.pro\files` folder and be named as shown.

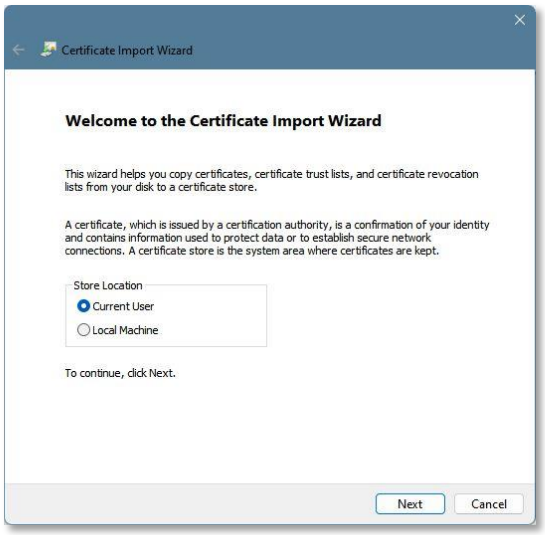
This will allow your device to authenticate to the WeeWX device and Maria DB server.

Installing Client Certificate on your Windows Device

The MyWX Pro Windows app is setup to allow the server to authenticate the Windows app. To do this you must install the Client certificate (i.e., the `client-cert.p12`) to your Windows devices on which the MyWX Pro app is installed. This will allow your device to authenticate to the WeeWX device and Maria DB server.

- ➢ Copy the `client-cert.p12` file to your Windows device.

To install to your Current User certificate store, right click on the file and select “Install PFX”. This starts the certificate install wizard. Follow the steps shown below, ensuring that the certificate is installed to the “Personal” store as shown.


Certificate Import Wizard

Welcome to the Certificate Import Wizard

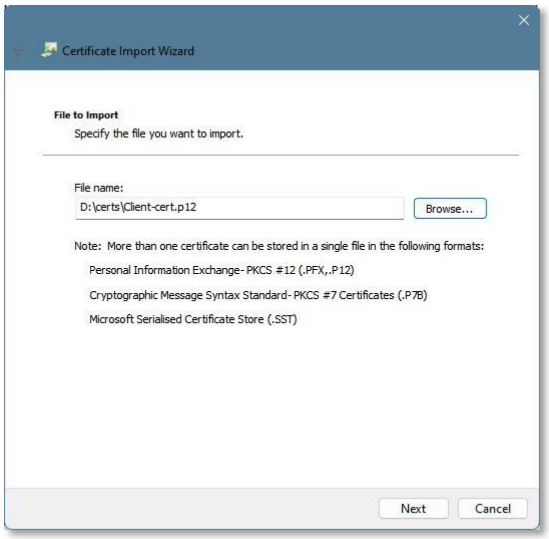
This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

Store Location
[✓ Current User]
Local Machine

To continue, click Next.

Figure 19 – Install to Current User



Certificate Import Wizard

File to Import

Specify the file you want to import.

File name:

D:\certs\Client-cert.p12

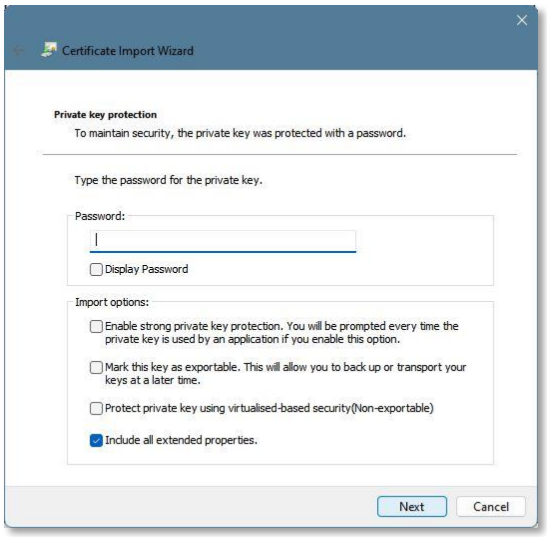
Note: More than one certificate can be stored in a single file in the following formats:

Personal Information Exchange- PKCS #12 (.PFX,.P12)

Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)

Microsoft Serialised Certificate Store (.SST)

Figure 20 – The certificate file to be installed



Certificate Import Wizard

Private key protection

To maintain security, the private key was protected with a password.

Type the password for the private key.

Password:

☐ Display Password

Import options:

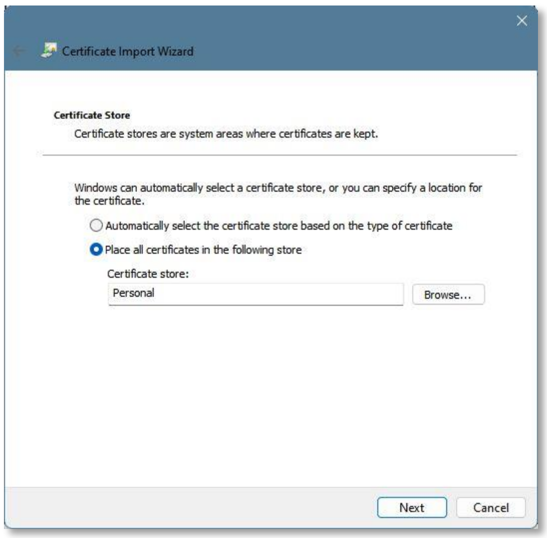
☐ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.

☐ Mark this key as exportable. This will allow you to back up or transport your keys at a later time.

☐ Protect private key using virtualised-based security(Non-exportable)

☑ Include all extended properties.

Figure 21 – Do not set a password



Certificate Store

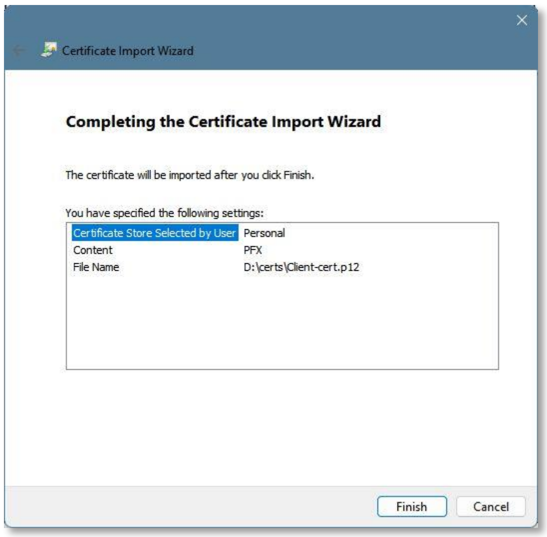
Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for the certificate.

Automatically select the certificate store based on the type of certificate [✓ Place all certificates in the following store]

Certificate store: Personal

Figure 22 – Import into the "Personal" store



Completing the Certificate Import Wizard

The certificate will be imported after you click Finish.

You have specified the following settings:

Certificate Store Selected by User	Personal
Content	PFX
File Name	D:\certs\Client-cert.p12

Figure 23 – Finish the install

Define Port Forwarding to WeeWX

A final step to make your WeeWX PWS accessible from the Internet is forwarding Port 3306 (i.e., the default port that Maria DB uses) to your Pi. Two things are required:

1. A fixed IP address for your Raspberry Pi;
2. A port forwarding definition in your Internet (ISP) router / modem. It specifies that port 3306 be forwarded to the Pi's fixed IP address.

How to define a port forward is very hardware specific. Consult the documentation for your ISP's router / modem.

List of Figures

Reference Documentation

Follow these links:

- https://github.com/David-Enst/WeeWX-BCRobotics/
- http://www.steves-internet-guide.com/ssl-certificates-explained/
- https://www.cyberciti.biz/faq/how-to-setup-mariadb-ssl-and-secure-connections-from-clients/
- https://stackoverflow.com/questions/4461360/how-to-install-trusted-ca-certificate-on-android-device
- https://www.lastbreach.com/blog/importing-private-ca-certificates-in-android
- https://mariadb.com/kb/en/using-tls-ssl-with-mariadb-java-connector/
- https://mariadb.com/kb/en/about-mariadb-connector-j/
- https://domoticproject.com/accessing-raspberry-ddns/ (Accessing from the Internet)