Advantech AE Technical Share Document

Date	2019/1/4	**SR#**	1-3643162399
Category	■FAQ □SOP	**Related OS**	N/A
Abstract	How to use MQTT TLS with iRTU device		
Keyword	MQTT, SSL, TLS, CA, certification, encrypted		
Related Product	ADAM-3600, ECU-1152, ECU-1251		

- **Problem Description:**

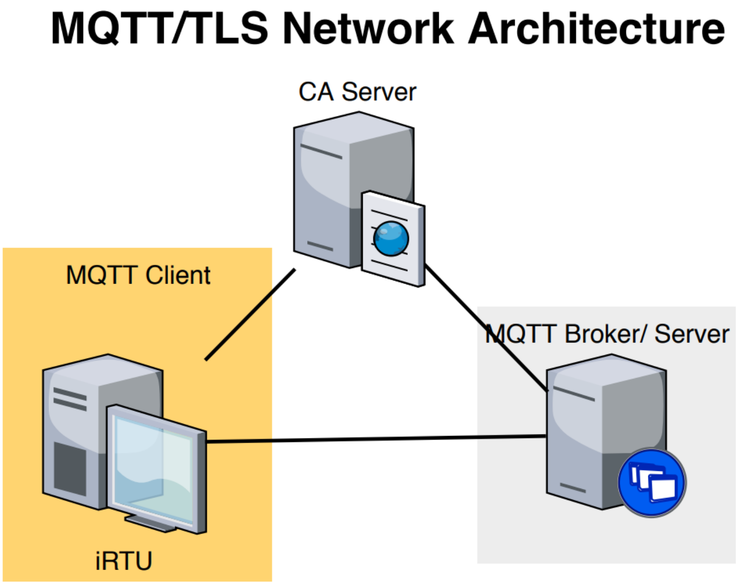
  User could use more security connection through SSL (Secure Sockets Layer)/ TLS (Transport Layer Security) setting. This document explains how to set up iRTU device with MQTT SSL/TLS.

- **Answer:**

  In the following steps, we introduce how to
  (1) Prepare Open SSL tool.
  (2) Generate CA key pair.
  (3) Generate a MQTT broker/ server key and certificate.
  (4) Configure Mosquitto MQTT Broker.
  (5) Upload data with iRTU (ADAM-3600) device.

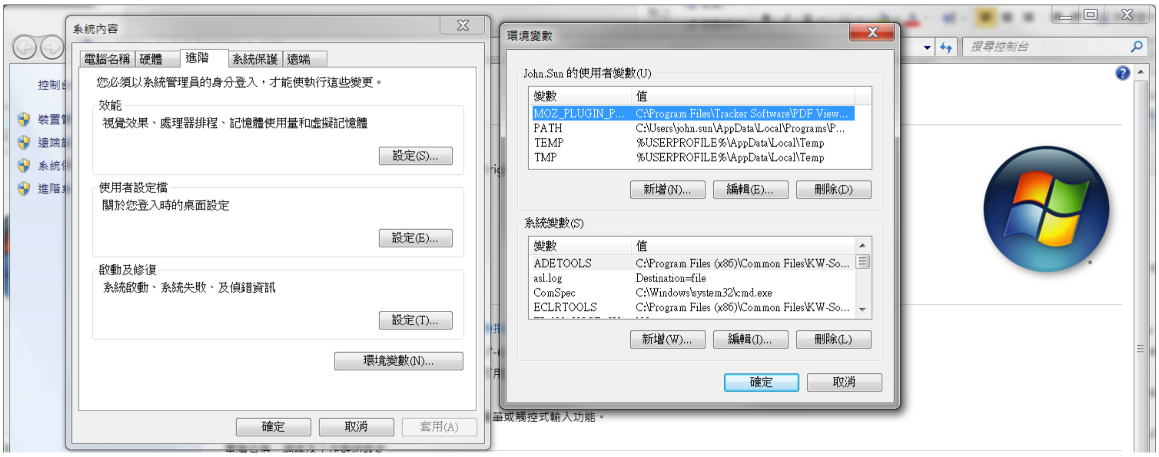
We also provide some tools and methods for debugging. User may base on his own needs to operate and test.

We would indicate the 3 roles (CA server, MQTT Broker, MQTT Client) of their setup.



1. Prepare OpenSSL tool

If user doesn’t have OpenSSL tool in his simulated CA server (or MQTT broker which may create certification), user needs download and install OpenSSL based on the computer environment (Open SSL 32bit/ 64bit).



系統內容

電腦名稱 | 硬體 | 進階 | 系統保護 | 遠端

您必須以系統管理員的身分登入，才能使執行這些變更。

效能
視覺效果、處理器排程、記憶體使用量和虛擬記憶體

使用者設定檔
關於您登入時的桌面設定

啟動及修復
系統啟動、系統失敗、及偵錯資訊

環境變數

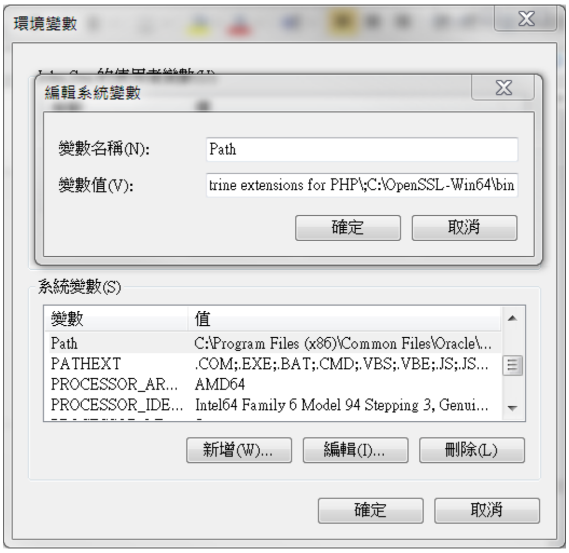
John.Sun 的使用者變數(U)

變數	值
MOZ_PLUGIN_P...	C:\Program Files\Tracker Software\PDF View...
PATH	C:\Users\john.sun\AppData\Local\Programs\P...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

系統變數(S)

變數	值
ADETOOLS	C:\Program Files (x86)\Common Files\KW-So...
asl.log	Destination=file
ComSpec	C:\Windows\system32\cmd.exe
ECLRTOOLS	C:\Program Files (x86)\Common Files\KW-So...

Add “**; C:\OpenSSL-Win64\bin**” (the path of installed OpenSSL) after the variable of Path.



環境變數

編輯系統變數

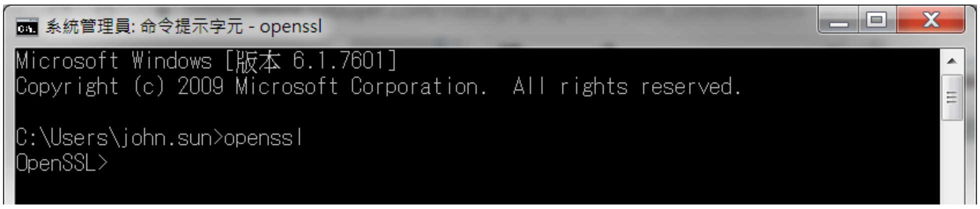
變數名稱(N): Path

變數值(V): trine extensions for PHP\;C:\OpenSSL-Win64\bin

系統變數(S)

變數	值
Path	C:\Program Files (x86)\Common Files\Oracle\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JS...
PROCESSOR_AR...	AMD64
PROCESSOR_IDE...	Intel64 Family 6 Model 94 Stepping 3, Genui...

After pressing “Confirm” to close the windows, user may open command line to test if “**openssl**” command could use.



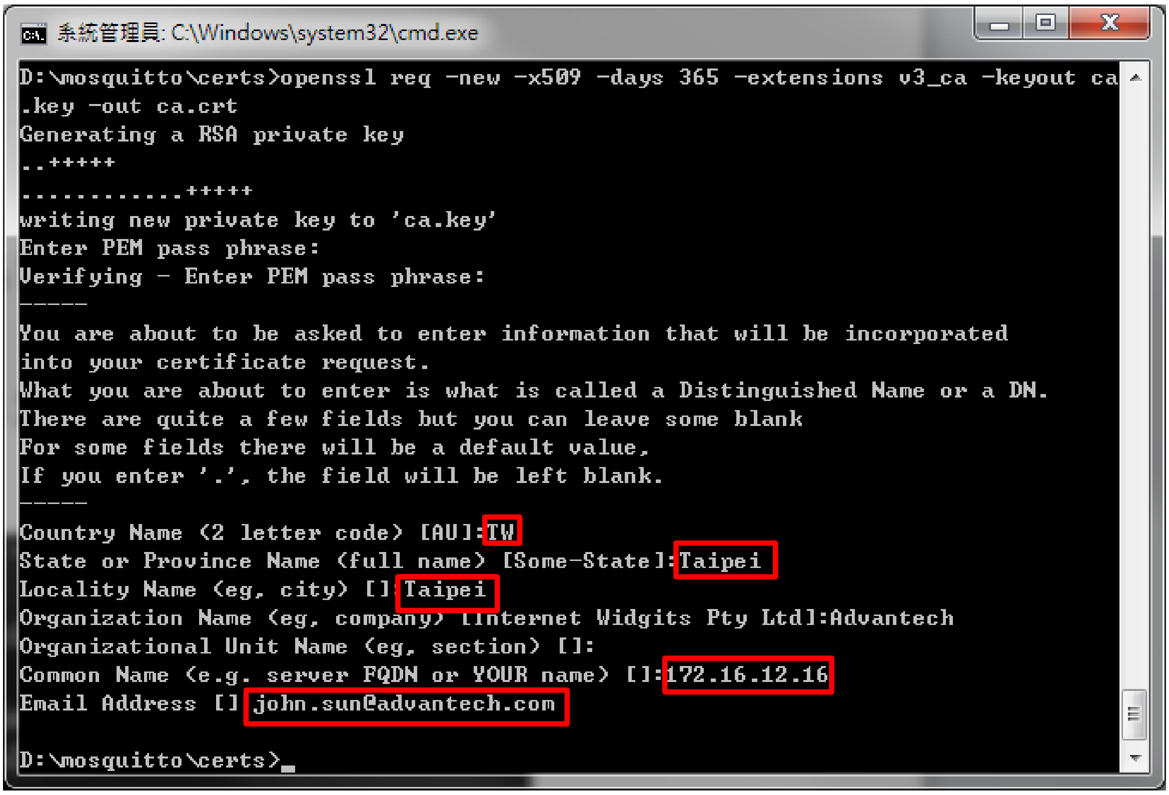
系統管理員: 命令提示字元 - openssl

```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\john.sun>openssl
OpenSSL>
```

2. Create a CA key pair (ca.key and ca.crt) in CA Server

Command is: **openssl req -new -x509 -days 365 -extensions v3_ca -keyout ca.key -out ca.crt**

Enter the simulated CA server information.



```
D:\mosquitto\certs>openssl req -new -x509 -days 365 -extensions v3_ca -keyout ca
.key -out ca.crt
Generating a RSA private key
..+++++
............+++++
writing new private key to 'ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TW
State or Province Name (full name) [Some-State]:Taipei
Locality Name (eg, city) []:Taipei
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Advantech
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:172.16.12.16
Email Address []:john.sun@advantech.com

D:\mosquitto\certs>_
```

During generating Certificate CA, user needs to fill in necessary information. Please note that Fully Qualified Domain Name is required in "Common Name", which is also acceptable to use IP address as Common Name.

3. Generate a MQTT broker/ server key and certificate.

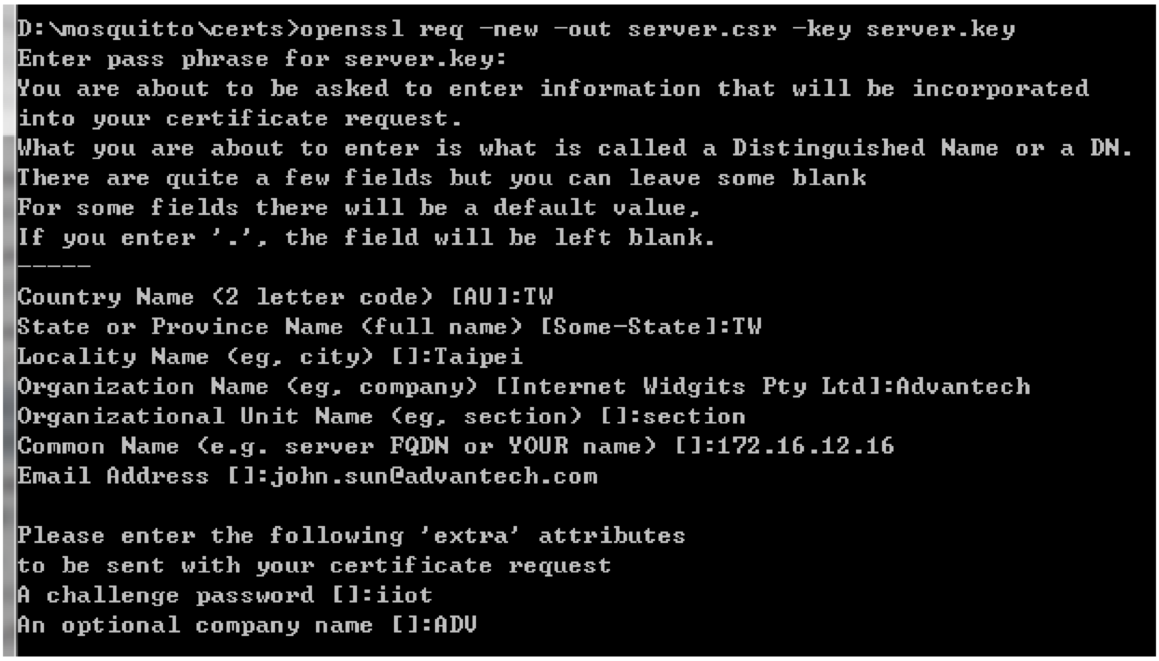
3.1 Generate a certificate signing request and key.

Now we create a server key pair which would be used by the broker.

User may create it in his MQTT broker/ server. In our following demo, the MQTT broker is the same server as CA server.

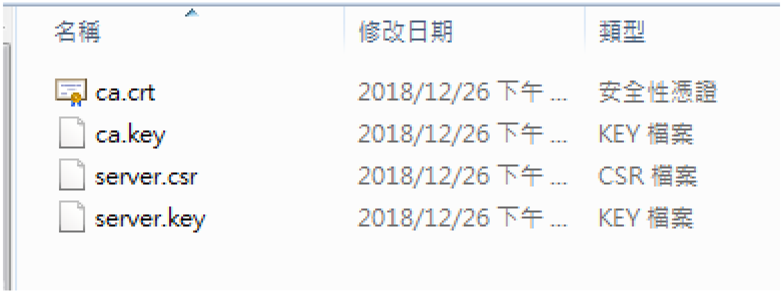
Command is: **openssl req -new -out server.csr -key server.key**

Enter server’s information as private key. It would generate server.csr (the intermediate file for generating server.crt) and server.key.



```
D:\mosquitto\certs>openssl req -new -out server.csr -key server.key
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TW
State or Province Name (full name) [Some-State]:TW
Locality Name (eg, city) []:Taipei
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Advantech
Organizational Unit Name (eg, section) []:section
Common Name (e.g. server FQDN or YOUR name) []:172.16.12.16
Email Address []:john.sun@advantech.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:iiot
An optional company name []:ADV
```

Since our CA server and MQTT broker are in the same machine, so far the files we created are listed as below.



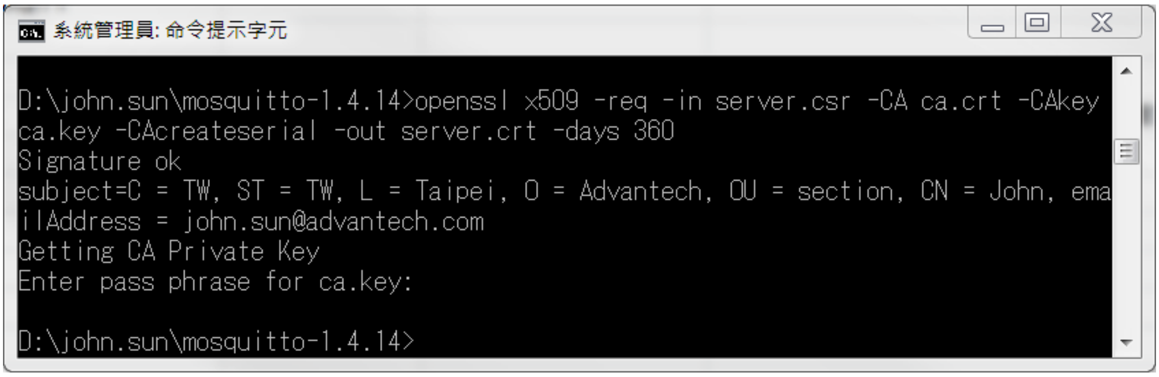
名稱	修改日期	類型
ca.crt	2018/12/26 下午 ...	安全性憑證
ca.key	2018/12/26 下午 ...	KEY 檔案
server.csr	2018/12/26 下午 ...	CSR 檔案
server.key	2018/12/26 下午 ...	KEY 檔案

3.3 Create “server.crt” file.

Now we use the **CA key** to verify and sign the server certificate (CSR). This creates the **server.crt** file.

If user has purchased the CA certificate, user would send the CSR file to CA server and get one CRT file. However, since we use the same machine as MQTT broker and CA server, we would import “**ca.crt**” during generating server.crt file.

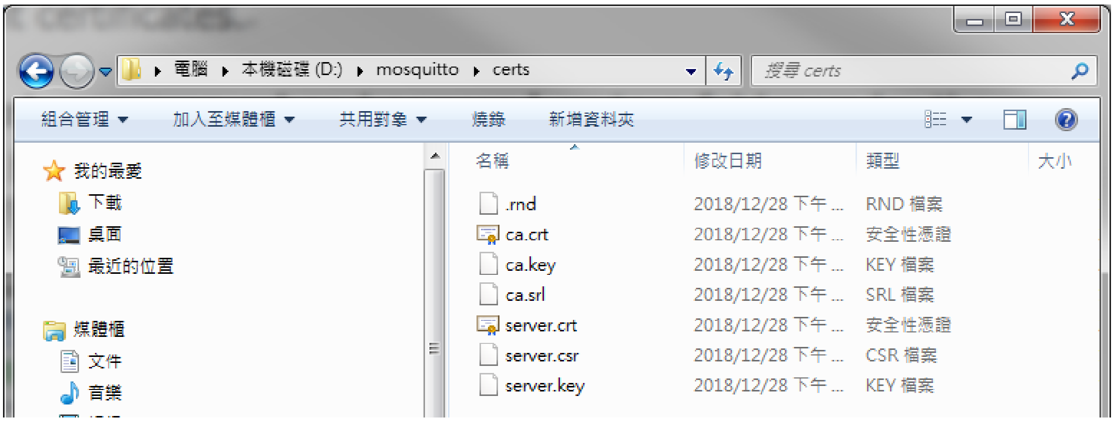
Command is: **openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 365**



系統管理員: 命令提示字元

```
D:\john.sun\mosquitto-1.4.14>openssl x509 -req -in server.csr -CA ca.crt -CAkey
ca.key -CAcreateserial -out server.crt -days 360
Signature ok
subject=C = TW, ST = TW, L = Taipei, O = Advantech, OU = section, CN = John, ema
ilAddress = john.sun@advantech.com
Getting CA Private Key
Enter pass phrase for ca.key:

D:\john.sun\mosquitto-1.4.14>
```

When prompted for the CN (Common Name), please enter either your server (or broker) hostname or domain name.



電腦 ▸ 本機磁碟 (D:) ▸ mosquitto ▸ certs

名稱	修改日期	類型	大小
.rnd	2018/12/28 下午 ...	RND 檔案	
ca.crt	2018/12/28 下午 ...	安全性憑證	
ca.key	2018/12/28 下午 ...	KEY 檔案	
ca.srl	2018/12/28 下午 ...	SRL 檔案	
server.crt	2018/12/28 下午 ...	安全性憑證	
server.csr	2018/12/28 下午 ...	CSR 檔案	
server.key	2018/12/28 下午 ...	KEY 檔案	

To prevent some issues caused during generating certificates and keys, user may use the attached generated key pair for testing.

tls.rar

4. Configure Mosquitto MQTT Broker.

4.1 Download and install Mosquitto MQTT Broker. https://mosquitto.org/download/

It is suggested to use default path “C:\Program Files\mosquitto” to prevent some environmental issue.

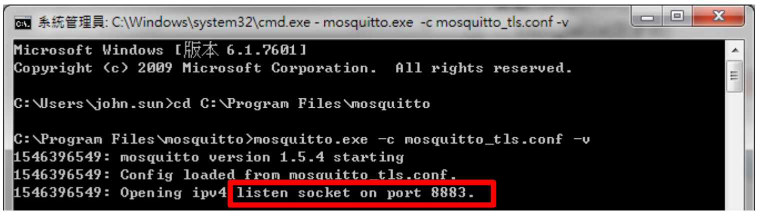
4.2 Copy “mosquito.conf” as “mosquito_tls.conf” and modify as below.

```
# Port to use for the default listener.
port 8883
# At least one of cafile or capath must be defined. They both
# define methods of accessing the PEM encoded Certificate
# Authority certificates that have signed your server certificate
# and that you wish to trust.
# cafile defines the path to a file containing the CA certificates.
# capath defines a directory that will be searched for files
# containing the CA certificates. For capath to work correctly, the
# certificate files must have ".crt" as the file ending and you must run
# "c_rehash <path to capath>" each time you add/remove a certificate.
cafile tls/ca.crt
#capath

# Path to the PEM encoded server certificate.
certfile tls/server.crt

# Path to the PEM encoded keyfile.
keyfile tls/server.key

# This option defines the version of the TLS protocol to use for this listener.
# The default value allows v1.2, v1.1 and v1.0, if they are all supported by
# the version of openssl that the broker was compiled against. For openssl >=
# 1.0.1 the valid values are tlsv1.2 tlsv1.1 and tlsv1. For openssl < 1.0.1 the
# valid values are tlsv1.
tls_version tlsv1.2
```

User needs to indicate the location of these 3 files (ca.crt, server.crt, server.key) for MQTT server. Because iRTU uses TLS version 1.2, we modify to indicate server to use tlsv1.2.

4.3 Start Mosquitto MQTT Broker

Command is: mosquito.exe –c mosquito_tls.conf –v



系統管理員: C:\Windows\system32\cmd.exe - mosquitto.exe -c mosquitto_tls.conf -v

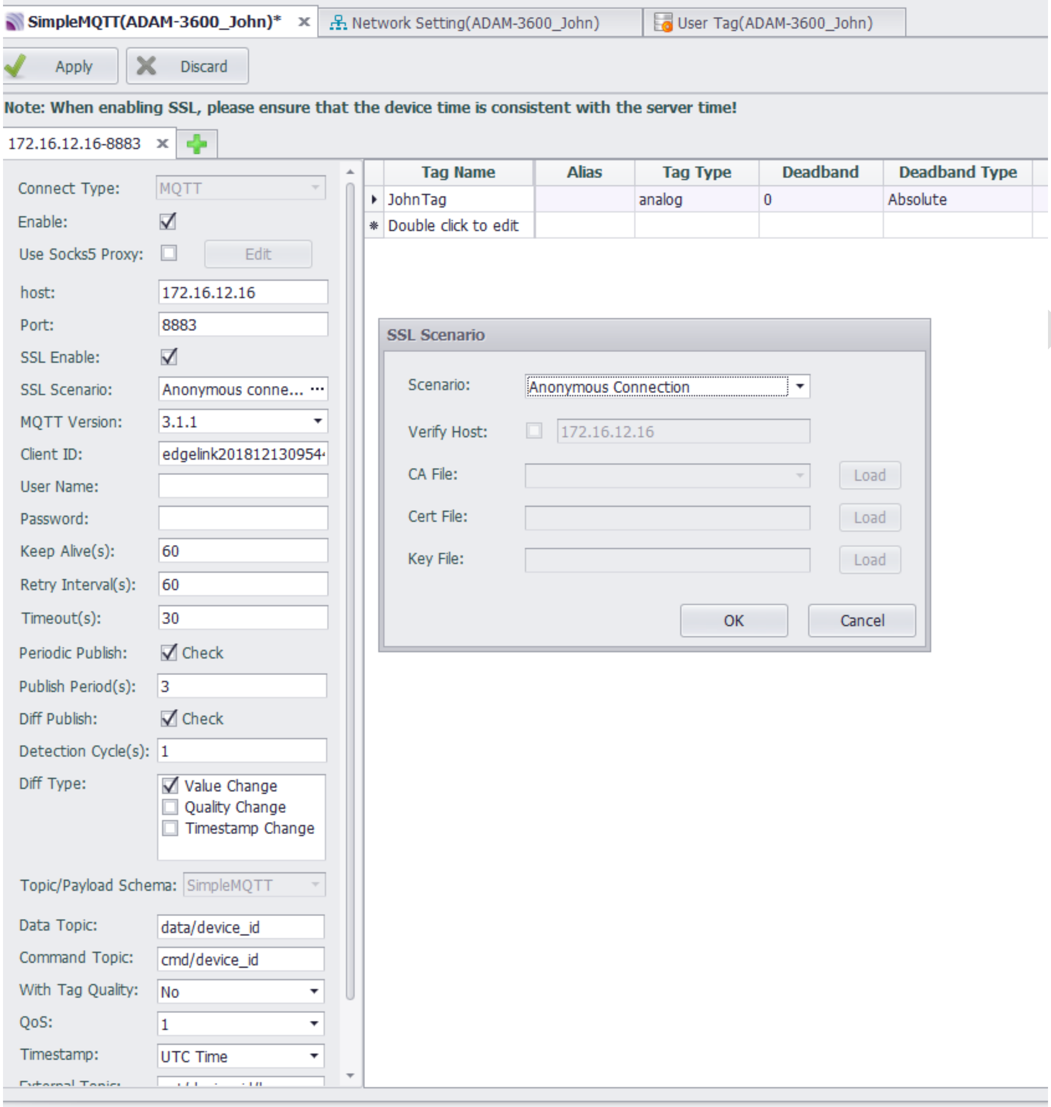
```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\john.sun>cd C:\Program Files\mosquitto

C:\Program Files\mosquitto>mosquitto.exe -c mosquitto_tls.conf -v
1546396549: mosquitto version 1.5.4 starting
1546396549: Config loaded from mosquitto_tls.conf.
1546396549: Opening ipv4 listen socket on port 8883.
```

5. Upload data with iRTU (ADAM-3600) device.

5.1 Use “Anonymous Connection”

In EdgeLink project, choose SimpleMQTT Cloud.

Select SSL Enable, and choose Anonymous Connection.



Note: When enabling SSL, please ensure that the device time is consistent with the server time!

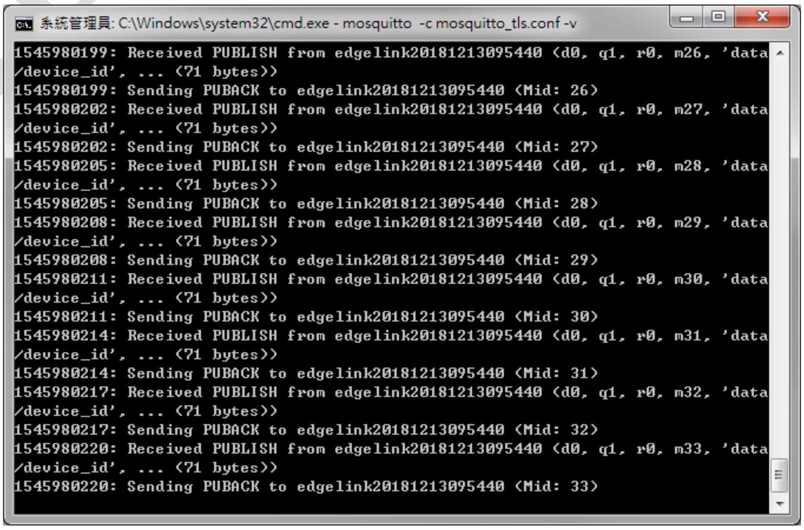
Connect Type:	MQTT
Enable:	☑
Use Socks5 Proxy:	☐
host:	172.16.12.16
Port:	8883
SSL Enable:	☑
SSL Scenario:	Anonymous conne...
MQTT Version:	3.1.1
Client ID:	edgelink201812130954
User Name:	
Password:	
Keep Alive(s):	60
Retry Interval(s):	60
Timeout(s):	30
Periodic Publish:	☑ Check
Publish Period(s):	3
Diff Publish:	☑ Check
Detection Cycle(s):	1
Diff Type:	☑ Value Change ☐ Quality Change ☐ Timestamp Change
Topic/Payload Schema:	SimpleMQTT
Data Topic:	data/device_id
Command Topic:	cmd/device_id
With Tag Quality:	No
QoS:	1
Timestamp:	UTC Time

Tag Name	Alias	Tag Type	Deadband	Deadband Type
JohnTag		analog	0	Absolute
Double click to edit				

SSL Scenario

Scenario:	Anonymous Connection
Verify Host:	☐ 172.16.12.16
CA File:	
Cert File:	
Key File:	

ADAM-3600 can use anonymous to connect and publish.

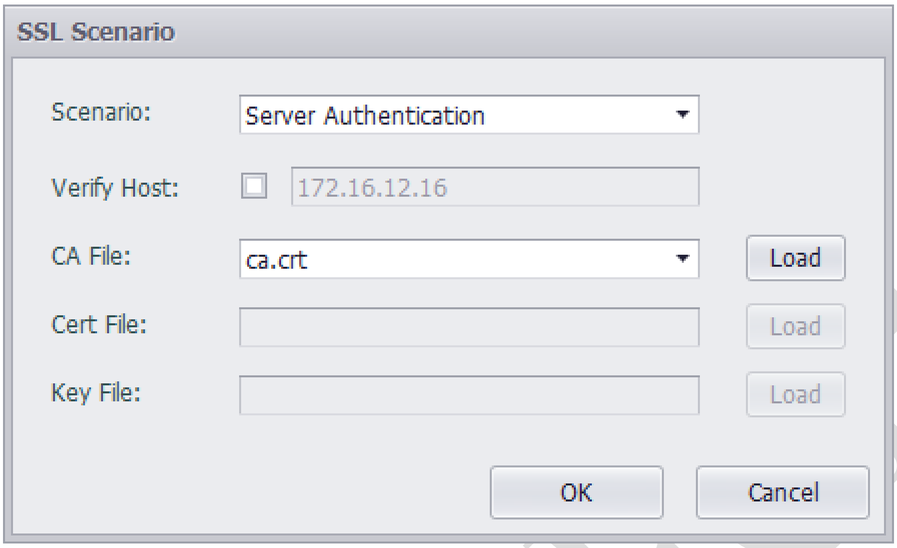


系統管理員: C:\Windows\system32\cmd.exe - mosquitto -c mosquitto_tls.conf -v

```
1545980199: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m26, 'data
/device_id', ... (71 bytes))
1545980199: Sending PUBACK to edgelink20181213095440 (Mid: 26)
1545980202: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m27, 'data
/device_id', ... (71 bytes))
1545980202: Sending PUBACK to edgelink20181213095440 (Mid: 27)
1545980205: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m28, 'data
/device_id', ... (71 bytes))
1545980205: Sending PUBACK to edgelink20181213095440 (Mid: 28)
1545980208: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m29, 'data
/device_id', ... (71 bytes))
1545980208: Sending PUBACK to edgelink20181213095440 (Mid: 29)
1545980211: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m30, 'data
/device_id', ... (71 bytes))
1545980211: Sending PUBACK to edgelink20181213095440 (Mid: 30)
1545980214: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m31, 'data
/device_id', ... (71 bytes))
1545980214: Sending PUBACK to edgelink20181213095440 (Mid: 31)
1545980217: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m32, 'data
/device_id', ... (71 bytes))
1545980217: Sending PUBACK to edgelink20181213095440 (Mid: 32)
1545980220: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m33, 'data
/device_id', ... (71 bytes))
1545980220: Sending PUBACK to edgelink20181213095440 (Mid: 33)
```

5.2 Use “Server Authentication” in iRTU device.

If MQTT broker uses self-signed certification, load the “ca.crt” which is used for creating server.key.



SSL Scenario

Scenario: Server Authentication

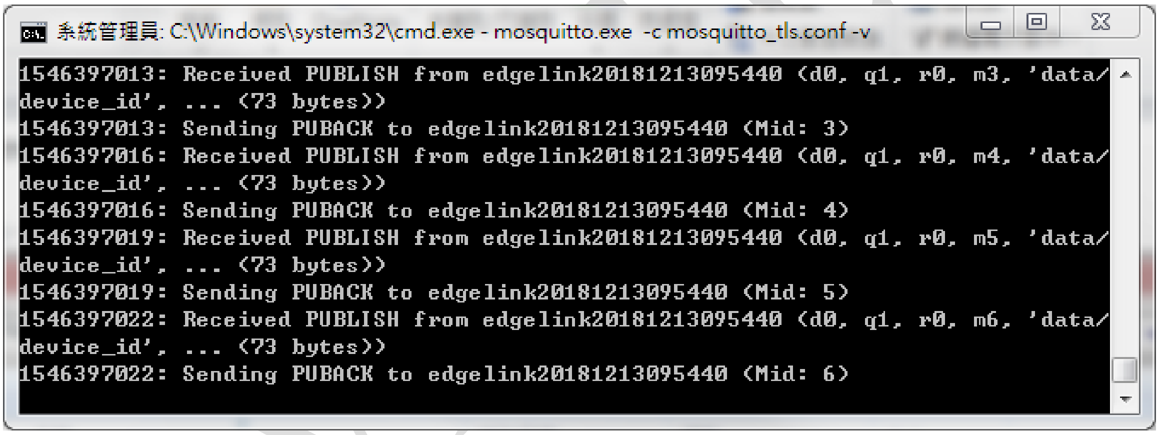
Verify Host: □ 172.16.12.16

CA File: ca.crt

Cert File:

Key File:

The result of successfully published message to Mosquitto TLS broker is shown as below.

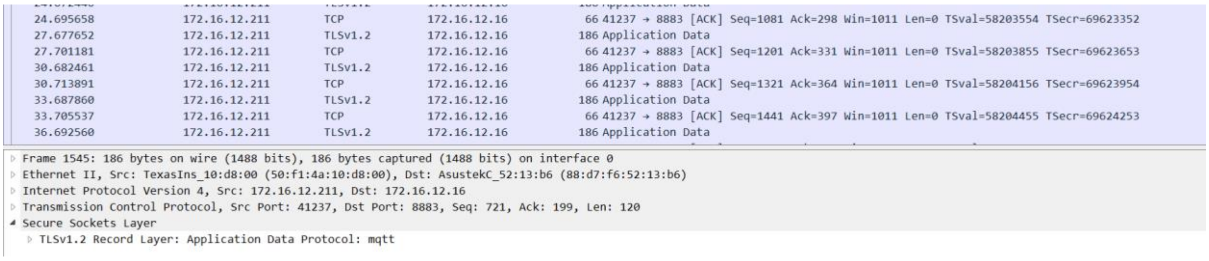


系統管理員: C:\Windows\system32\cmd.exe - mosquitto.exe -c mosquitto_tls.conf -v

```
1546397013: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m3, 'data/
device_id', ... (73 bytes))
1546397013: Sending PUBACK to edgelink20181213095440 (Mid: 3)
1546397016: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m4, 'data/
device_id', ... (73 bytes))
1546397016: Sending PUBACK to edgelink20181213095440 (Mid: 4)
1546397019: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m5, 'data/
device_id', ... (73 bytes))
1546397019: Sending PUBACK to edgelink20181213095440 (Mid: 5)
1546397022: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m6, 'data/
device_id', ... (73 bytes))
1546397022: Sending PUBACK to edgelink20181213095440 (Mid: 6)
```

6. Other tools for debugging

6.1 **Wireshark** can verify if it is using TLS MQTT protocol.

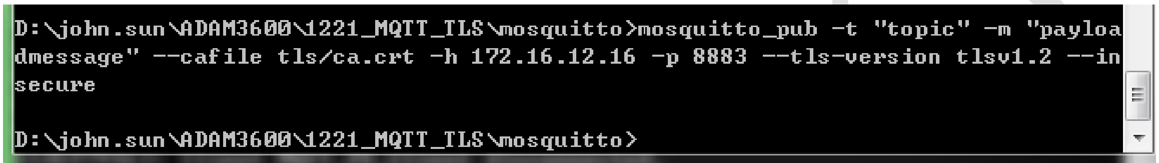


```
24.695658      172.16.12.211     TCP       172.16.12.16       66 41237 → 8883 [ACK] Seq=1081 Ack=298 Win=1011 Len=0 TSval=58203554 TSecr=69623352
27.677652      172.16.12.211     TLSv1.2   172.16.12.16      186 Application Data
27.701181      172.16.12.211     TCP       172.16.12.16       66 41237 → 8883 [ACK] Seq=1201 Ack=331 Win=1011 Len=0 TSval=58203855 TSecr=69623653
30.682461      172.16.12.211     TLSv1.2   172.16.12.16      186 Application Data
30.713891      172.16.12.211     TCP       172.16.12.16       66 41237 → 8883 [ACK] Seq=1321 Ack=364 Win=1011 Len=0 TSval=58204156 TSecr=69623954
33.687860      172.16.12.211     TLSv1.2   172.16.12.16      186 Application Data
33.705537      172.16.12.211     TCP       172.16.12.16       66 41237 → 8883 [ACK] Seq=1441 Ack=397 Win=1011 Len=0 TSval=58204455 TSecr=69624253
36.692560      172.16.12.211     TLSv1.2   172.16.12.16      186 Application Data

▷ Frame 1545: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface 0
▷ Ethernet II, Src: TexasIns_10:d8:00 (50:f1:4a:10:d8:00), Dst: AsustekC_52:13:b6 (88:d7:f6:52:13:b6)
▷ Internet Protocol Version 4, Src: 172.16.12.211, Dst: 172.16.12.16
▷ Transmission Control Protocol, Src Port: 41237, Dst Port: 8883, Seq: 721, Ack: 199, Len: 120
◢ Secure Sockets Layer
  ▷ TLSv1.2 Record Layer: Application Data Protocol: mqtt
```

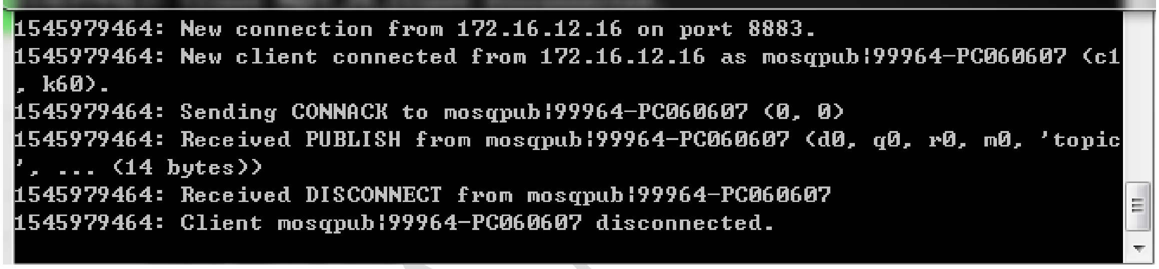
6.2 The command of "**mosquitto_pub**"

To check if the server is working, user could load the same "ca.crt" used in the iRTU device.

Command is: **mosquito_pub –t "topic" –m "payloadmessage" –cafile tls/ca.crt –h 172.16.12.16 –p 8883 –tls-version tlsv1.2 --insecure**

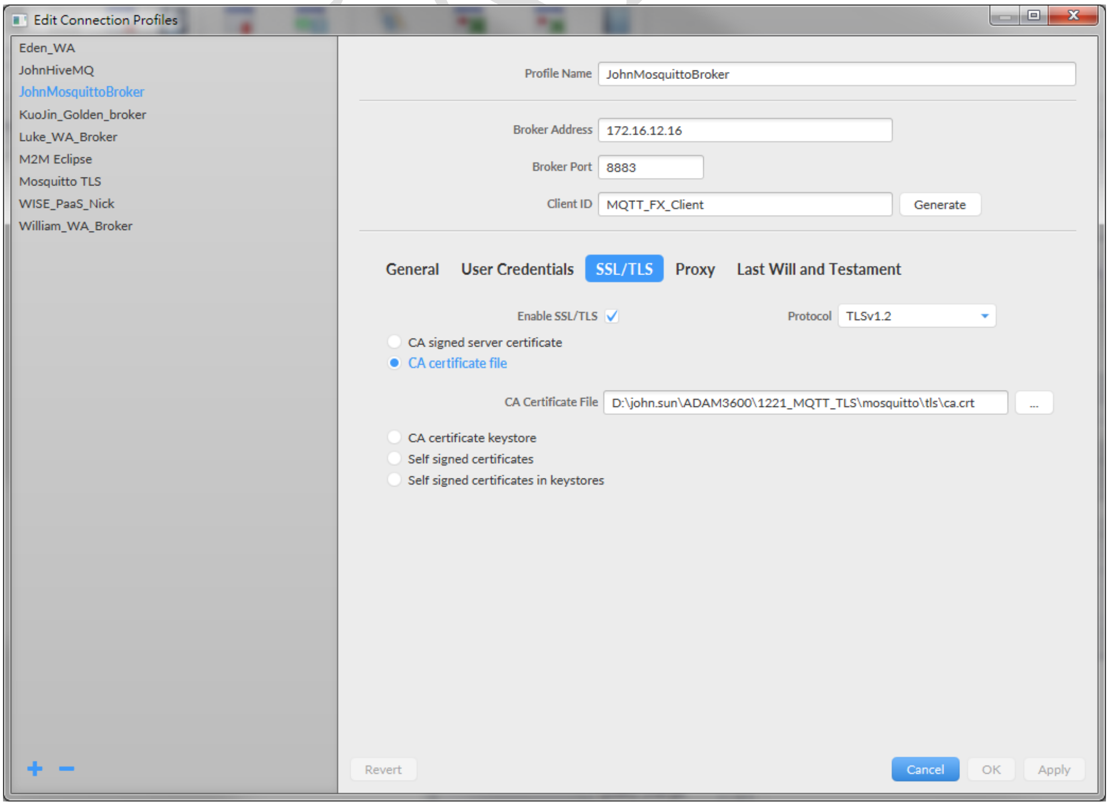


```
D:\john.sun\ADAM3600\1221_MQTT_TLS\mosquitto>mosquitto_pub -t "topic" -m "payloa
dmessage" --cafile tls/ca.crt -h 172.16.12.16 -p 8883 --tls-version tlsv1.2 --in
secure

D:\john.sun\ADAM3600\1221_MQTT_TLS\mosquitto>
```



```
1545979464: New connection from 172.16.12.16 on port 8883.
1545979464: New client connected from 172.16.12.16 as mosqpub|99964-PC060607 (c1
, k60).
1545979464: Sending CONNACK to mosqpub|99964-PC060607 (0, 0)
1545979464: Received PUBLISH from mosqpub|99964-PC060607 (d0, q0, r0, m0, 'topic
', ... (14 bytes))
1545979464: Received DISCONNECT from mosqpub|99964-PC060607
1545979464: Client mosqpub|99964-PC060607 disconnected.
```

6.3 Use other 3rd **party MQTT Client software** to load the "ca.crt" file to connect.



Edit Connection Profiles

Eden_WA
JohnHiveMQ
JohnMosquittoBroker
KuoJin_Golden_broker
Luke_WA_Broker
M2M Eclipse
Mosquitto TLS
WISE_PaaS_Nick
William_WA_Broker

Profile Name: JohnMosquittoBroker

Broker Address: 172.16.12.16

Broker Port: 8883

Client ID: MQTT_FX_Client

Enable SSL/TLS ✓

Protocol: TLSv1.2

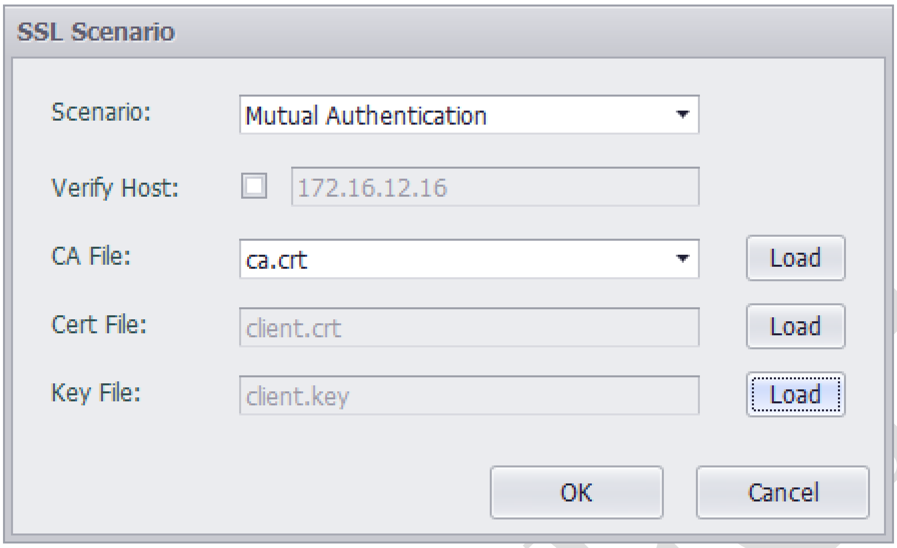
○ CA signed server certificate
● CA certificate file

CA Certificate File: D:\john.sun\ADAM3600\1221_MQTT_TLS\mosquitto\tls\ca.crt

○ CA certificate keystore
○ Self signed certificates
○ Self signed certificates in keystores

7. Use "Mutual Authentication"

For the 3rd scenario user could choose in EdgeLink is mutual authentication, which enable the MQTT broker to verify the connecting client certificate.



SSL Scenario

Scenario: Mutual Authentication

Verify Host: □ 172.16.12.16

CA File: ca.crt

Cert File: client.crt

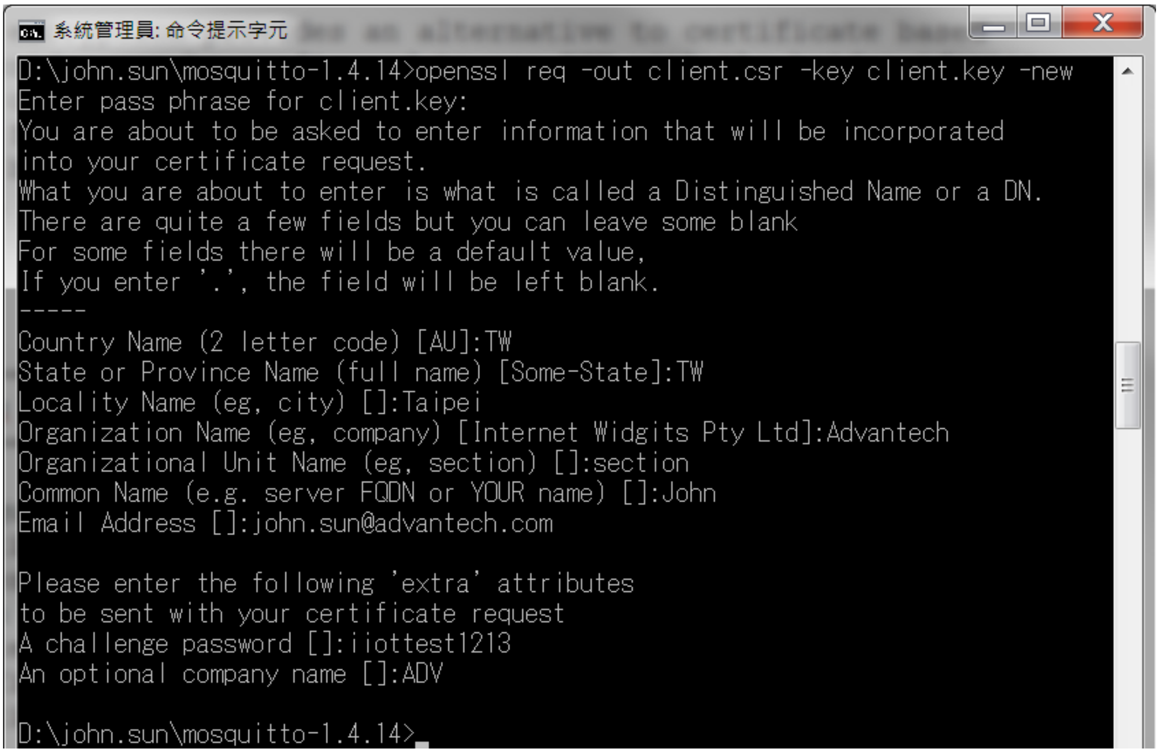
Key File: client.key

We introduce the steps to use this scenario as below.

7.1 Generate a client key/ certificate pair.

User may generate a certificate signing request and key in MQTT client side. However, in our demo, we use CA server to generate the client.key and client.crt.

Command is: **openssl req -out client.csr -key client.key –new**



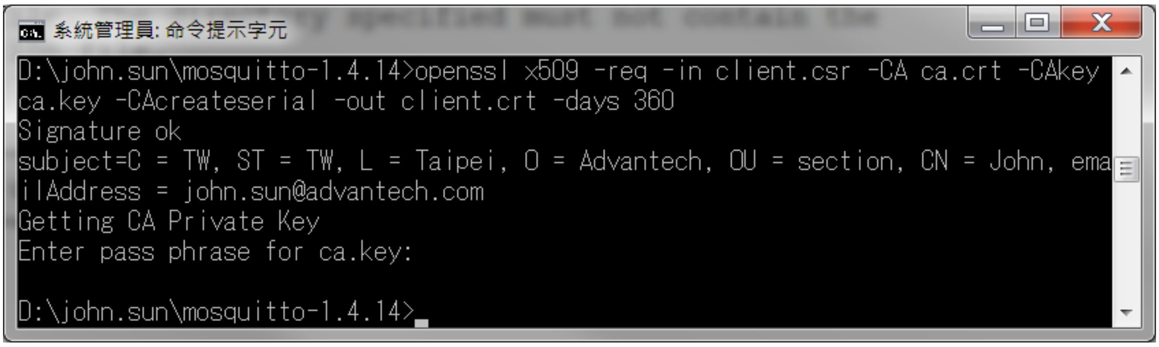
系統管理員: 命令提示字元

```
D:\john.sun\mosquitto-1.4.14>openssl req -out client.csr -key client.key -new
Enter pass phrase for client.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TW
State or Province Name (full name) [Some-State]:TW
Locality Name (eg, city) []:Taipei
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Advantech
Organizational Unit Name (eg, section) []:section
Common Name (e.g. server FQDN or YOUR name) []:John
Email Address []:john.sun@advantech.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:iiottest1213
An optional company name []:ADV

D:\john.sun\mosquitto-1.4.14>_
```

7.2 Generate “client.crt” file.

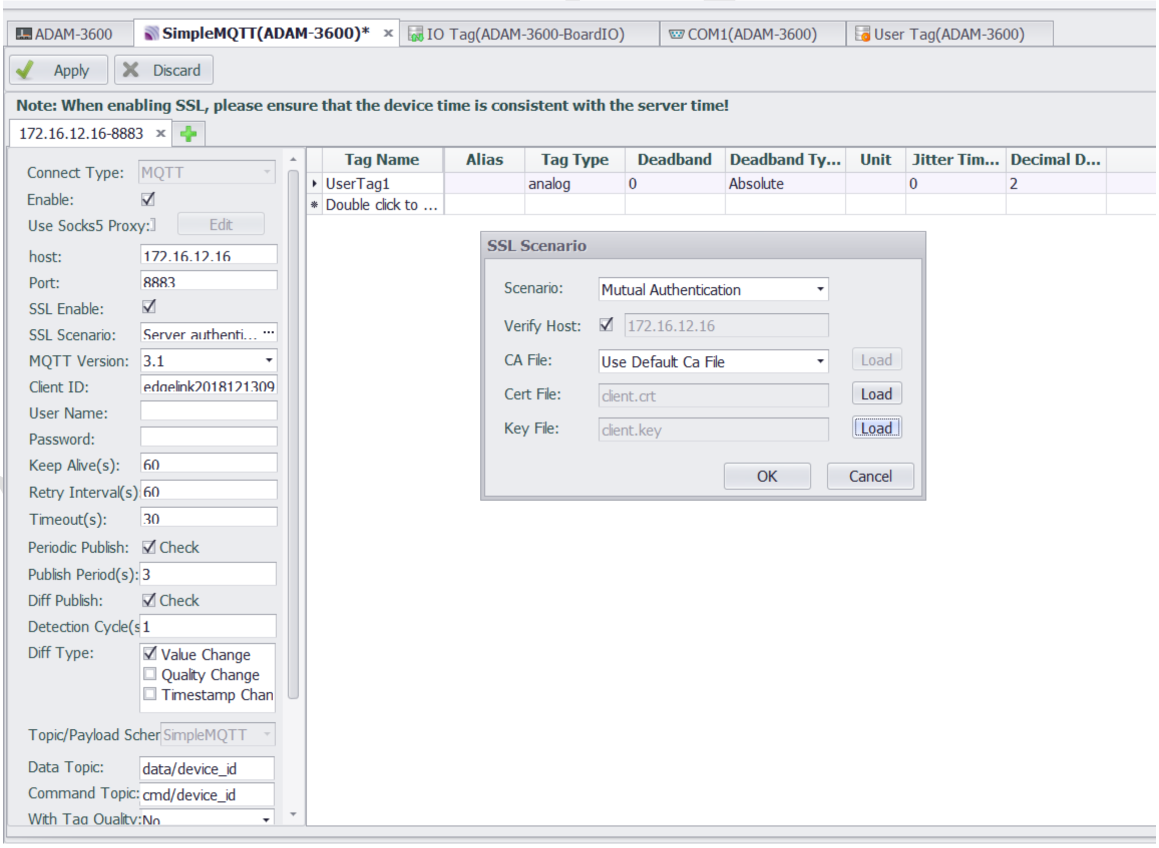
Now we use the **CA key** to verify and sign the server certificate (CSR). This creates the **client.crt** file.

Command is: **openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out client.crt -days <duration>**



```
D:\john.sun\mosquitto-1.4.14>openssl x509 -req -in client.csr -CA ca.crt -CAkey
ca.key -CAcreateserial -out client.crt -days 360
Signature ok
subject=C = TW, ST = TW, L = Taipei, O = Advantech, OU = section, CN = John, ema
ilAddress = john.sun@advantech.com
Getting CA Private Key
Enter pass phrase for ca.key:

D:\john.sun\mosquitto-1.4.14>_
```

7.3 Load the **CA certificate file (client.crt, client.key, ca.crt)** to the client.



Note: When enabling SSL, please ensure that the device time is consistent with the server time!

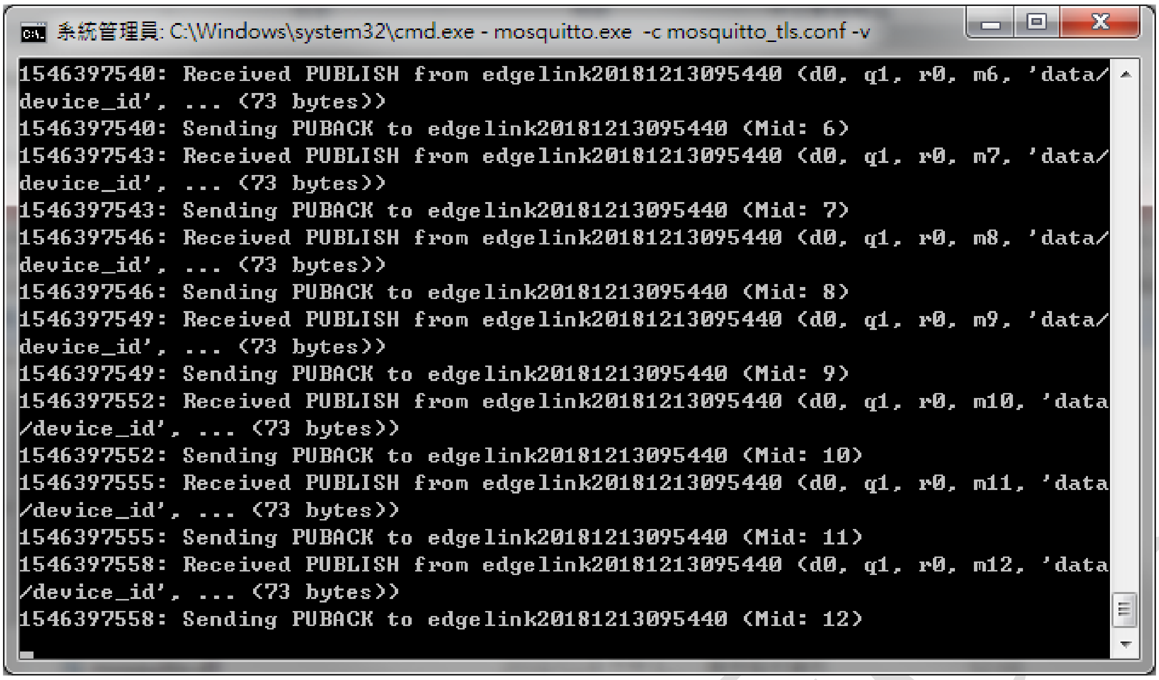
Connect Type: MQTT
Enable: ☑
Use Socks5 Proxy: □
host: 172.16.12.16
Port: 8883
SSL Enable: ☑
SSL Scenario: Server authenti...
MQTT Version: 3.1
Client ID: edgelink2018121309
User Name:
Password:
Keep Alive(s): 60
Retry Interval(s) 60
Timeout(s): 30
Periodic Publish: ☑ Check
Publish Period(s): 3
Diff Publish: ☑ Check
Detection Cycle(s 1
Diff Type: ☑ Value Change □ Quality Change □ Timestamp Chan
Topic/Payload Scher SimpleMQTT
Data Topic: data/device_id
Command Topic: cmd/device_id
With Tag Quality: No

Tag Name	Alias	Tag Type	Deadband	Deadband Ty...	Unit	Jitter Tim...	Decimal D...
UserTag1		analog	0	Absolute		0	2
Double click to ...							

SSL Scenario

Scenario: Mutual Authentication
Verify Host: ☑ 172.16.12.16
CA File: Use Default Ca File
Cert File: client.crt
Key File: client.key

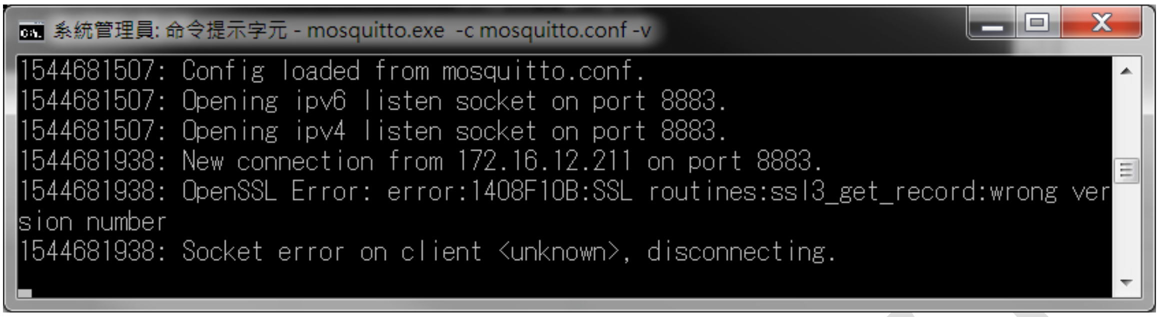
User could see the uploading results on Mosquitto broker.


系統管理員: C:\Windows\system32\cmd.exe - mosquitto.exe -c mosquitto_tls.conf -v

```
1546397540: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m6, 'data/
device_id', ... (73 bytes))
1546397540: Sending PUBACK to edgelink20181213095440 (Mid: 6)
1546397543: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m7, 'data/
device_id', ... (73 bytes))
1546397543: Sending PUBACK to edgelink20181213095440 (Mid: 7)
1546397546: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m8, 'data/
device_id', ... (73 bytes))
1546397546: Sending PUBACK to edgelink20181213095440 (Mid: 8)
1546397549: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m9, 'data/
device_id', ... (73 bytes))
1546397549: Sending PUBACK to edgelink20181213095440 (Mid: 9)
1546397552: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m10, 'data
/device_id', ... (73 bytes))
1546397552: Sending PUBACK to edgelink20181213095440 (Mid: 10)
1546397555: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m11, 'data
/device_id', ... (73 bytes))
1546397555: Sending PUBACK to edgelink20181213095440 (Mid: 11)
1546397558: Received PUBLISH from edgelink20181213095440 (d0, q1, r0, m12, 'data
/device_id', ... (73 bytes))
1546397558: Sending PUBACK to edgelink20181213095440 (Mid: 12)
```

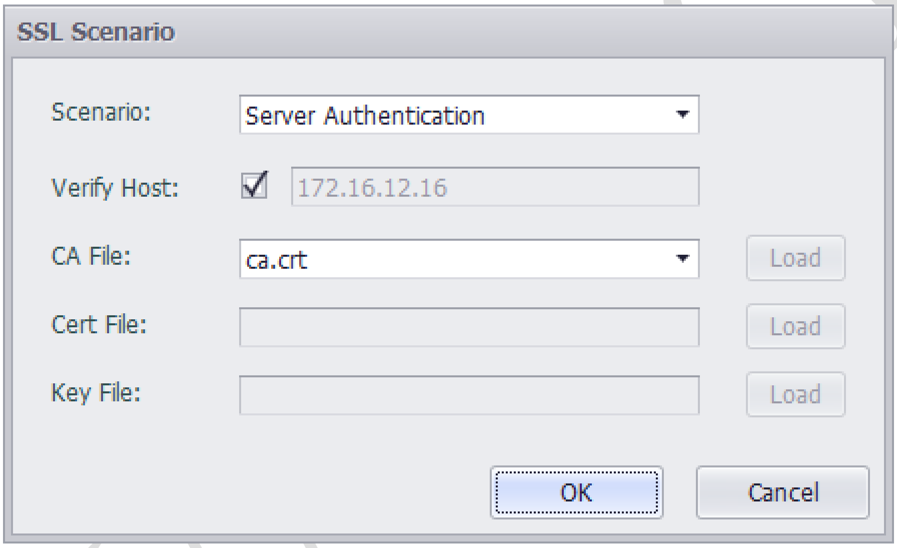
8. Trouble shooting

8.1 Because iRTU device uses TLS Version 1.2, in mosquitto.conf, it needs to be configured as "tls_version tlsv1.2".


系統管理員: 命令提示字元 - mosquitto.exe -c mosquitto.conf -v

```
1544681507: Config loaded from mosquitto.conf.
1544681507: Opening ipv6 listen socket on port 8883.
1544681507: Opening ipv4 listen socket on port 8883.
1544681938: New connection from 172.16.12.211 on port 8883.
1544681938: OpenSSL Error: error:1408F10B:SSL routines:ssl3_get_record:wrong ver
sion number
1544681938: Socket error on client <unknown>, disconnecting.
```

8.2 Because our broker is not verified host by CA, we cannot use "Verify Host" to connect.



Scenario: Server Authentication

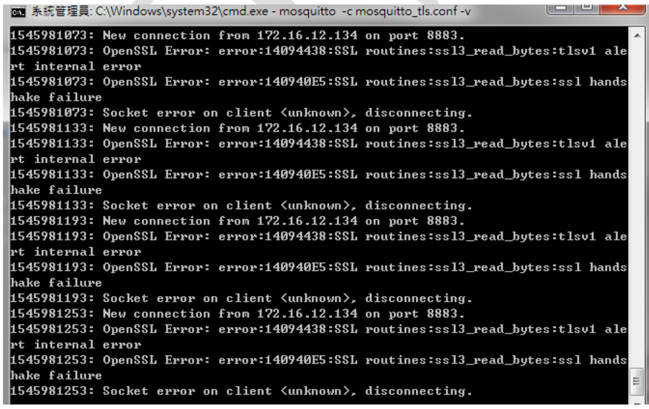
Verify Host: ☑ 172.16.12.16

CA File: ca.crt

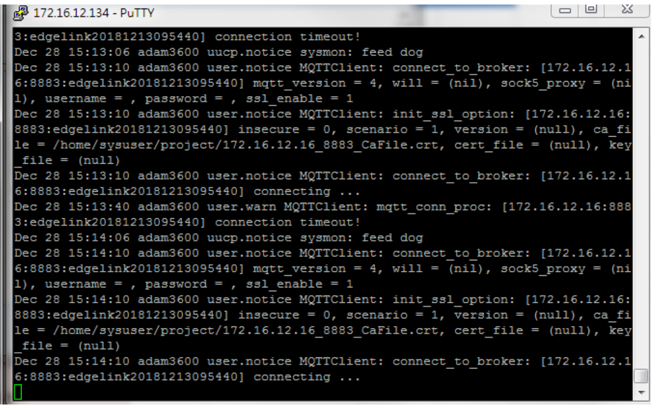
Cert File:

Key File:

It would not successfully connect to the MQTT broker if the broker does not purchase certificate.


系統管理員: C:\Windows\system32\cmd.exe - mosquitto -c mosquitto_tls.conf -v

```
1545981073: New connection from 172.16.12.134 on port 8883.
1545981073: OpenSSL Error: error:14094438:SSL routines:ssl3_read_bytes:tlsv1 ale
rt internal error
1545981073: OpenSSL Error: error:140940E5:SSL routines:ssl3_read_bytes:ssl hands
hake failure
1545981073: Socket error on client <unknown>, disconnecting.
1545981133: New connection from 172.16.12.134 on port 8883.
1545981133: OpenSSL Error: error:14094438:SSL routines:ssl3_read_bytes:tlsv1 ale
rt internal error
1545981133: OpenSSL Error: error:140940E5:SSL routines:ssl3_read_bytes:ssl hands
hake failure
1545981133: Socket error on client <unknown>, disconnecting.
1545981193: New connection from 172.16.12.134 on port 8883.
1545981193: OpenSSL Error: error:14094438:SSL routines:ssl3_read_bytes:tlsv1 ale
rt internal error
1545981193: OpenSSL Error: error:140940E5:SSL routines:ssl3_read_bytes:ssl hands
hake failure
1545981193: Socket error on client <unknown>, disconnecting.
1545981253: New connection from 172.16.12.134 on port 8883.
1545981253: OpenSSL Error: error:14094438:SSL routines:ssl3_read_bytes:tlsv1 ale
rt internal error
1545981253: OpenSSL Error: error:140940E5:SSL routines:ssl3_read_bytes:ssl hands
hake failure
1545981253: Socket error on client <unknown>, disconnecting.
```



```
3:edgelink20181213095440] connection timeout!
Dec 28 15:13:06 adam3600 uucp.notice sysmon: feed dog
Dec 28 15:13:10 adam3600 user.notice MQTTClient: connect_to_broker: [172.16.12.1
6:8883:edgelink20181213095440] mqtt_version = 4, will = (nil), sock5_proxy = (ni
l), username = , password = , ssl_enable = 1
Dec 28 15:13:10 adam3600 user.notice MQTTClient: init_ssl_option: [172.16.12.16:
8883:edgelink20181213095440] insecure = 0, scenario = 1, version = (null), ca_fi
le = /home/sysuser/project/172.16.12.16_8883_CaFile.crt, cert_file = (null), key
_file = (null)
Dec 28 15:13:10 adam3600 user.notice MQTTClient: connect_to_broker: [172.16.12.1
6:8883:edgelink20181213095440] connecting ...
Dec 28 15:13:40 adam3600 user.warn MQTTClient: mqtt_conn_proc: [172.16.12.16:888
3:edgelink20181213095440] connection timeout!
Dec 28 15:14:06 adam3600 uucp.notice sysmon: feed dog
Dec 28 15:14:10 adam3600 user.notice MQTTClient: connect_to_broker: [172.16.12.1
6:8883:edgelink20181213095440] mqtt_version = 4, will = (nil), sock5_proxy = (ni
l), username = , password = , ssl_enable = 1
Dec 28 15:14:10 adam3600 user.notice MQTTClient: init_ssl_option: [172.16.12.16:
8883:edgelink20181213095440] insecure = 0, scenario = 1, version = (null), ca_fi
le = /home/sysuser/project/172.16.12.16_8883_CaFile.crt, cert_file = (null), key
_file = (null)
Dec 28 15:14:10 adam3600 user.notice MQTTClient: connect_to_broker: [172.16.12.1
6:8883:edgelink20181213095440] connecting ...
```