<https://medium.com/@kajsuaning/mqtt-mutual-certificate-authentication-f51bc6e1a457>

**1.Install openssl**

sudo apt-get update

sudo apt-get install openssl

**2.Create a private key for our certificate authority.**

openssl genrsa -des3 -out ca.key 2048

**3.Create a certificate for our certificate authority.**

openssl req -new -x509 -days 1826 -key ca.key -out ca.crt

Country Name (2 letter code) [AU]: AU

State or Province Name (full name) [Some-State]:

Locality Name (eg, city) []:

Organization Name (eg, company) [Internet Widgits Pty Ltd]:

Organizational Unit Name (eg, section) []:

Common Name (e.g. server FQDN or YOUR name) []: myCA

Email Address []:

password:password

**4.Generate a server private key.**

openssl genrsa -out server.key 2048

**5.Generate a certificate signing request (csr) for the server.**

openssl req -new -out server.csr -key server.key

Country Name (2 letter code) [AU]: AU

State or Province Name (full name) [Some-State]:

Locality Name (eg, city) []:

Organization Name (eg, company) [Internet Widgits Pty Ltd]:

Organizational Unit Name (eg, section) []:

Common Name (e.g. server FQDN or YOUR name) []: localhost

Email Address []:

**6. Validate the server certificate signing request with the certificate authority (which is us)**

openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 360

**7. Generate a client private key.**

openssl genrsa -out esp32.key 2048

**8. Generate a certificate signing request (csr) for the client.**

openssl req -new -out esp32.csr -key esp32.key

Country Name (2 letter code) [AU]: AU

State or Province Name (full name) [Some-State]:

Locality Name (eg, city) []:

Organization Name (eg, company) [Internet Widgits Pty Ltd]:

Organizational Unit Name (eg, section) []:

Common Name (e.g. server FQDN or YOUR name) []: myclient

Email Address []:

**9. Validate the client certificate signing request with the certificate authority (which is us).**

openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out esp32.crt -days 360

**Mosquitto MQTT Broker Configuration**

**1.Install Mosquitto MQTT Broker.**

sudo apt update

sudo apt install -y mosquitto

sudo systemctl status mosquitto

**2. Copy the certificate authority certificate (ca.crt), server certificate (server.crt) & server private key (server.key) to /etc/mosquitto/certs on your server.**

**3. Update the Mosquitto configuration to support mutual certificate authentication.**

Create a custom configuration file:

sudo nano /etc/mosquitto/conf.d/custom.conf

# Don't allow unauthenticated users

allow\_anonymous false

# Port to listen on

listener 8883

# Uses the common name from the client cert as the username

use\_identity\_as\_username true

# Path the the Certificate Authority Cert.

cafile /etc/mosquitto/certs/ca.crt

# Path to server private key

keyfile /etc/mosquitto/certs/server.key

# Path to server certificate

certfile /etc/mosquitto/certs/server.crt

# Set TLS version

tls\_version tlsv1.2

# Require that clients provide certificates

require\_certificate true

**4. Restart the Mosquitto service.**

sudo service mosquitto restart

sudo cp /root/mosquitto\_certs/\*.crt /etc/mosquitto/certs/

sudo cp /root/mosquitto\_certs/\*.key /etc/mosquitto/certs/

**#change the ownership of the files to the mosquitto user and group**

sudo chown mosquitto:mosquitto /etc/mosquitto/certs/\*

**#permissions are set properly**

sudo chmod 644 /etc/mosquitto/certs/\*

sudo chmod 600 /etc/mosquitto/certs/\*.key

**Validate Trust Chain**

openssl verify -CAfile ca.crt server.crt

openssl verify -CAfile ca.crt client.crt

openssl verify -CAfile /etc/mosquitto/certs/ca.crt /etc/mosquitto/certs/server.crt

openssl verify -CAfile /etc/mosquitto/certs/ca.crt /etc/mosquitto/certs/client.crt

sudo systemctl restart mosquitto

//after re-making cert files

sudo chmod 644 /etc/mosquitto/certs/\*

sudo chown mosquitto:mosquitto /etc/mosquitto/certs/\*

sudo chown mosquitto:mosquitto /etc/mosquitto/ca\_certificates/\*

sudo chown mosquitto:mosquitto /etc/mosquitto/aclfile/\*

**#test connection**

telnet 192.168.100.53 8883

mosquitto\_sub -h 192.168.100.53 -p 1883 -t "test/topic"

~~openssl s\_client -connect 192.168.100.53:8883 -CAfile /etc/mosquitto/certs/ca.crt -cert /etc/mosquitto/certs/client.crt -key /etc/mosquitto/certs/client.key~~

~~mosquitto\_sub -h 192.168.100.53 -p 8883 -t "test/topic" --cafile /etc/mosquitto/certs/ca.crt --cert /etc/mosquitto/certs/client.crt --key /etc/mosquitto/certs/client.key~~

~~mosquitto\_pub -h 192.168.100.53 -p 8883 -t "test/topic" -m "Hello from mTLS!" --cafile /etc/mosquitto/certs/ca.crt --cert /etc/mosquitto/certs/client.crt --key /etc/mosquitto/certs/client.key~~

//dupa ce conexiunea a fost realizata si functioneaza, vom folosi esp32-ul pentru a ne conecta la server, iar din program avem un output de genul:

Connected!

Message published

Message received on /test: Hello from ESP32

Message published

Message received on /test: Hello from ESP32

Message published

Message received on /test: Hello from ESP32

Message published

**Test:**

root@ubuntu-srv-24:/etc/mosquitto/certs# mosquitto\_sub -h 192.168.100.53 -p 8883 -t '/test' --cafile /etc/mosquitto/certs/ca.crt --cert /etc/mosquitto/certs/esp32.crt --key /etc/mosquitto/certs/esp32.key

Hello from ESP32

Hello from ESP32

**MQTT Logs: On Ubuntu, check Mosquitto logs:**

sudo tail -f /var/log/mosquitto/mosquitto.log

alte comenzi folositoare:

su - root

adduser username sudo

adduser root sudo

usermod -aG sudo user

sudo passwd root

sudo nano /etc/mosquitto/conf.d/mtls.conf

sudo systemctl status mosquitto.service

sudo systemctl restart mosquitto

sudo systemctl status mosquitto

mosquitto -c /etc/mosquitto/mosquitto.conf -v

sudo nano /etc/mosquitto/mosquitto.conf

ps -aux |grep mosquitto

sudo lsof -i :8883

sudo netstat -tulnp | grep 8883

sudo kill -9 <PID of apache>

t1: mosquitto\_sub -t '/test'

t2: mosquitto\_pub -t '/test' -m 'Hello world'