Ubuntu

Mosquitto MQTT TLS Debugging Guide

1️⃣ Initial Setup & Certificate Configuration

Generate TLS Certificates

```
sudo mkdir -p /etc/mosquitto/certs
cd /etc/mosquitto/certs
```

Generate CA Certificate

```
sudo openssl genrsa -out ca.key 2048
sudo openssl req -new -x509 -days 365 -key ca.key -out ca.crt -subj "/CN=MyCA"
```

Generate Server Certificate

```
sudo openssl genrsa -out mosquitto.key 2048
sudo openssl req -new -key mosquitto.key -out mosquitto.csr -subj
"/CN=<192.168.95.20_Broker_IP_Address>"
sudo openssl x509 -req -in mosquitto.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out
mosquitto.crt -days 365
```

Set Permissions

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

2️⃣ Configure Mosquitto for TLS

Edit the Mosquitto configuration file:

```
sudo nano /etc/mosquitto/mosquitto.conf
```

Add the following lines(mosquitto.conf):

```
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /var/run/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/
```

```
log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d


listener 1883
allow_anonymous true



# Listener for secure connections (SSL/TLS)
listener 8883

cafile /etc/mosquitto/certs/ca.crt
keyfile /etc/mosquitto/certs/server.key
certfile /etc/mosquitto/certs/server.crt
require_certificate false
# Enable TLS version 1.2
tls_version tlsv1.2
log_type all
log_dest stdout
# Optional: Allow only TLS v1.2 connections



# Other configurations...
log_type all
log_dest stdout
```

Restart Mosquitto:

```
sudo systemctl restart mosquitto
```

📡 Publish & Subscribe Messages Over TLS

Subscribe to a Topic

```
mosquitto_sub -h <Broker_IP_Address> -p 8883 --cafile /etc/mosquitto/certs/ca.crt -t "test/topic"
```

Publish a Message

```
mosquitto_pub -h <Broker_IP_Address> -p 8883 --cafile /etc/mosquitto/certs/ca.crt -t "test/topic" -m "Hello"
```

3️⃣ Debugging Mosquitto TLS Issues

🖥️ Run Mosquitto in Debug Mode

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

🔍 Check If Mosquitto Is Listening on Port 8883

```
sudo netstat -tulnp | grep mosquitto
```

🔗 Test TLS Connection with OpenSSL

```
sudo openssl s_client -connect localhost:8883 -CAfile /etc/mosquitto/certs/ca.crt
```

If this fails, check if the **CN** in the certificate matches the hostname:

```
openssl x509 -in /etc/mosquitto/certs/mosquitto.crt -text -noout | grep "Subject:"
```

If CN is incorrect, regenerate the certificate:

```
sudo openssl req -new -x509 -days 365 -key /etc/mosquitto/certs/mosquitto.key -out
/etc/mosquitto/certs/mosquitto.crt -subj "/CN=<Broker_IP_Address>"
```

🔀 Handling CN & IP Issues in TLS

Issue: Using IP Instead of CN Causes TLS Error

If your Mosquitto broker's certificate is issued to a specific Common Name (CN) (e.g., "mosquitto"), then using the IP instead of the CN can cause a TLS error.

1️⃣ Check CN in the Certificate

```
openssl x509 -in /etc/mosquitto/certs/mosquitto.crt -text -noout | grep "Subject:"
```

If it shows:

```
subject=CN = mosquitto
```

Then, using an IP (e.g., `172.18.8.195`) will fail TLS verification.

2️⃣ Solution: Use the CN Instead of the IP

Try using `mosquitto` instead of the IP:

```
mosquitto_sub -h mosquitto -p 8883 --cafile /etc/mosquitto/certs/ca.crt -t "test/topic"
```

If `mosquitto` is not resolvable, update `/etc/hosts`:

```
sudo nano /etc/hosts
```

Add this line (replace with actual IP):

```
172.18.8.195 mosquitto
```

Save and exit (`Ctrl + X`, `Y`, `Enter`). Now retry:

```
mosquitto_sub -h mosquitto -p 8883 --cafile /etc/mosquitto/certs/ca.crt -t "test/topic"
```

3️⃣ Alternative: Disable Hostname Verification (Temporary Fix)

```
mosquitto_sub -h 172.18.8.195 -p 8883 --cafile /etc/mosquitto/certs/ca.crt --insecure -t "test/topic"
```

🚨 This bypasses hostname verification, so only use it for testing.

4️⃣ Permanent Fix: Generate a Certificate with the Correct IP

If you must use the IP, regenerate the certificate with Subject Alternative Names (SANs):

```
sudo openssl req -new -key mosquitto.key -out mosquitto.csr -subj "/CN=mosquitto" -addext "subjectAltName = IP:172.18.8.195"
sudo openssl x509 -req -in mosquitto.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out mosquitto.crt -days 365
```

Restart Mosquitto:

```
sudo systemctl restart mosquitto
```

🛠️ Check Logs for Errors

```
sudo journalctl -u mosquitto -f
```

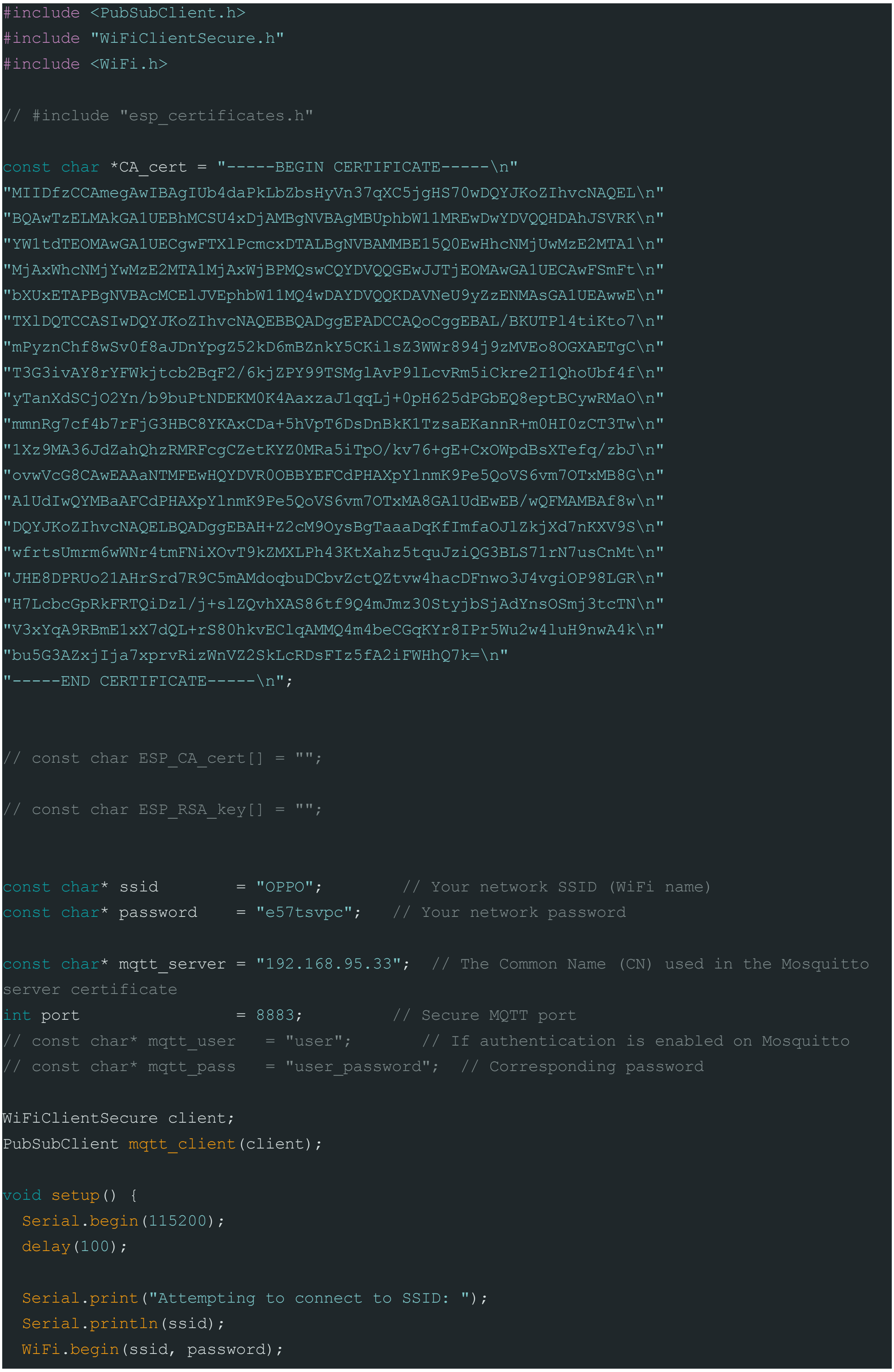
🔥 Summary of Fixes

Issue	Command/Fix

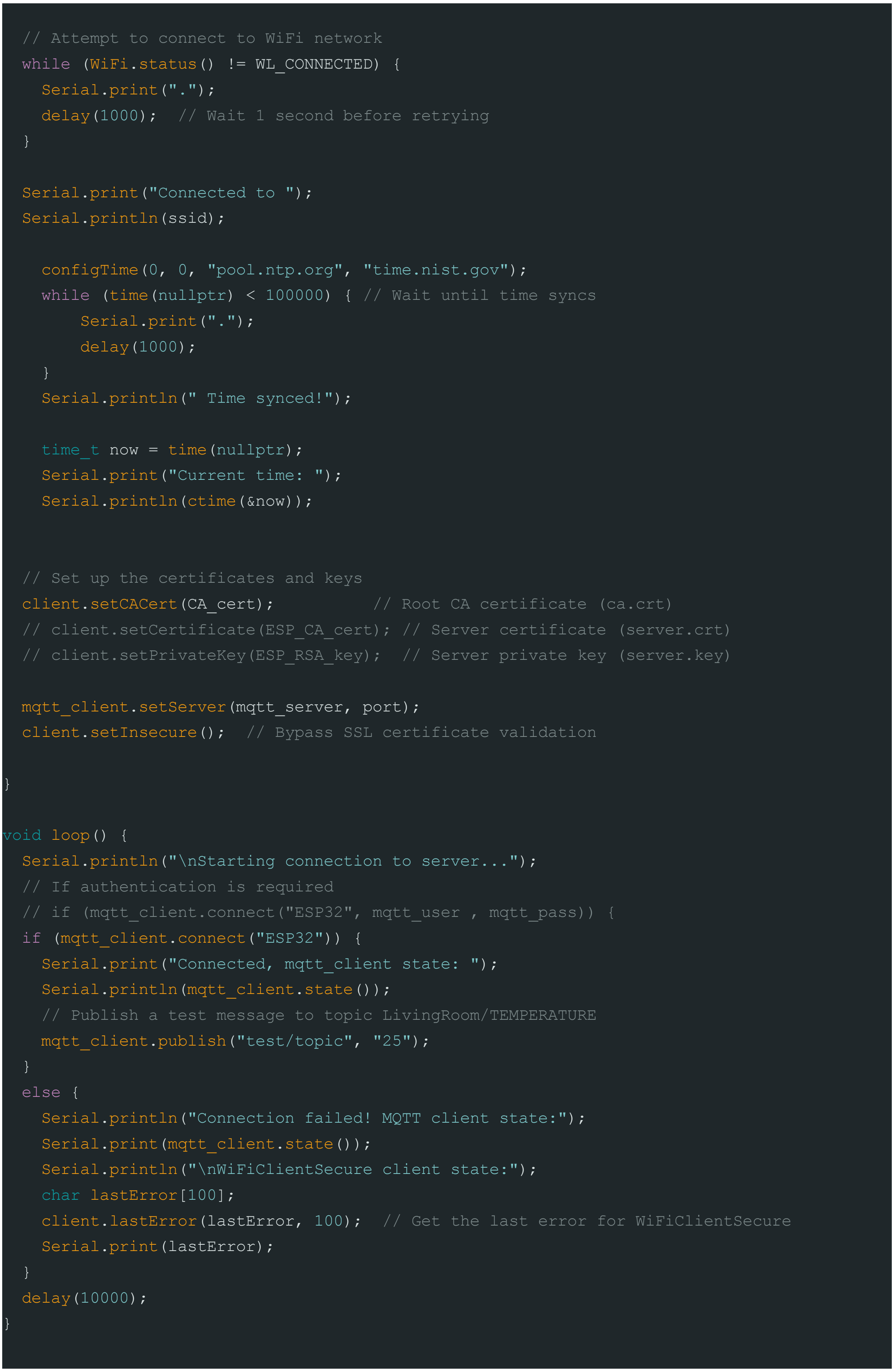
Mosquitto not listening on 8883	`sudo netstat -tulnp	grep mosquitto`
OpenSSL connection failure	sudo openssl s_client -connect localhost:8883 -CAfile /etc/mosquitto/certs/ca.crt	
Certificate CN mismatch	`openssl x509 -in /etc/mosquitto/certs/mosquitto.crt -text -noout	grep "Subject:"`
Regenerate certificates	sudo openssl req -new -x509 -days 365 -key /etc/mosquitto/certs/mosquitto.key -out /etc/mosquitto/certs/mosquitto.crt -subj "/CN=mosquitto"	
Restart Mosquitto	sudo systemctl restart mosquitto	
Debug logs	sudo journalctl -u mosquitto -f	
Run Mosquitto in verbose mode	sudo mosquitto -c /etc/mosquitto/mosquitto.conf -v	

This guide should help in debugging and resolving Mosquitto TLS issues across multiple VMs. 🚀

ESP32 Code



```
#include <PubSubClient.h>
#include "WiFiClientSecure.h"
#include <WiFi.h>

// #include "esp_certificates.h"

const char *CA_cert = "-----BEGIN CERTIFICATE-----\n"
"MIIDfzCCAmegAwIBAgIUb4daPkLbZbsHyVn37qXC5jgHS70wDQYJKoZIhvcNAQEL\n"
"BQAwTzELMAkGA1UEBhMCSU4xDjAMBgNVBAgMBUphbW11MREwDwYDVQQHDAhJSVRK\n"
"YW1tdTEOMAwGA1UECgwFTXlPcmcxDTALBgNVBAMMBE15Q0EwHhcNMjUwMzE2MTA1\n"
"MjAxWhcNMjYwMzE2MTA1MjAxWjBPMQswCQYDVQQGEwJJTjEOMAwGA1UECAwFSmFt\n"
"bXUxETAPBgNVBAcMCElJVEphbW11MQ4wDAYDVQQKDAVNeU9yZzENMAsGA1UEAwwE\n"
"TXlDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAL/BKUTPl4tiKto7\n"
"mPyznChf8wSv0f8aJDnYpgZ52kD6mBZnkY5CKilsZ3WWr894j9zMVEo8OGXAETgC\n"
"T3G3ivAY8rYFWkjtcb2BqF2/6kjZPY99TSMglAvP9lLcvRm5iCkre2I1QhoUbf4f\n"
"yTanXdSCjO2Yn/b9buPtNDEKM0K4AaxzaJ1qqLj+0pH625dPGbEQ8eptBCywRMaO\n"
"mmnRg7cf4b7rFjG3HBC8YKAxCDa+5hVpT6DsDnBkK1TzsaEKannR+m0HI0zCT3Tw\n"
"1Xz9MA36JdZahQhzRMRFcgCZetKYZ0MRa5iTpO/kv76+gE+CxOWpdBsXTefq/zbJ\n"
"ovwVcG8CAwEAAaNTMFEwHQYDVR0OBBYEFCdPHAXpYlnmK9Pe5QoVS6vm7OTxMB8G\n"
"A1UdIwQYMBaAFCdPHAXpYlnmK9Pe5QoVS6vm7OTxMA8GA1UdEwEB/wQFMAMBAf8w\n"
"DQYJKoZIhvcNAQELBQADggEBAH+Z2cM9OysBgTaaaDqKfImfaOJlZkjXd7nKXV9S\n"
"wfrtsUmrm6wWNr4tmFNiXOvT9kZMXLPh43KtXahz5tquJziQG3BLS71rN7usCnMt\n"
"JHE8DPRUo21AHrSrd7R9C5mAMdoqbuDCbvZctQZtvw4hacDFnwo3J4vgiOP98LGR\n"
"H7LcbcGpRkFRTQiDzl/j+slZQvhXAS86tf9Q4mJmz30StyjbSjAdYnsOSmj3tcTN\n"
"V3xYqA9RBmE1xX7dQL+rS80hkvEClqAMMQ4m4beCGqKYr8IPr5Wu2w4luH9nwA4k\n"
"bu5G3AZxjIja7xprvRizWnVZ2SkLcRDsFIz5fA2iFWHhQ7k=\n"
"-----END CERTIFICATE-----\n";


// const char ESP_CA_cert[] = "";

// const char ESP_RSA_key[] = "";


const char* ssid        = "OPPO";        // Your network SSID (WiFi name)
const char* password    = "e57tsvpc";   // Your network password

const char* mqtt_server = "192.168.95.33";  // The Common Name (CN) used in the Mosquitto
server certificate
int port                = 8883;         // Secure MQTT port
// const char* mqtt_user   = "user";       // If authentication is enabled on Mosquitto
// const char* mqtt_pass   = "user_password";  // Corresponding password

WiFiClientSecure client;
PubSubClient mqtt_client(client);

void setup() {
  Serial.begin(115200);
  delay(100);

  Serial.print("Attempting to connect to SSID: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
```



```
  // Attempt to connect to WiFi network
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(1000);  // Wait 1 second before retrying
  }

  Serial.print("Connected to ");
  Serial.println(ssid);

    configTime(0, 0, "pool.ntp.org", "time.nist.gov");
    while (time(nullptr) < 100000) { // Wait until time syncs
        Serial.print(".");
        delay(1000);
    }
    Serial.println(" Time synced!");

    time_t now = time(nullptr);
    Serial.print("Current time: ");
    Serial.println(ctime(&now));


  // Set up the certificates and keys
  client.setCACert(CA_cert);          // Root CA certificate (ca.crt)
  // client.setCertificate(ESP_CA_cert); // Server certificate (server.crt)
  // client.setPrivateKey(ESP_RSA_key);  // Server private key (server.key)

  mqtt_client.setServer(mqtt_server, port);
  client.setInsecure();  // Bypass SSL certificate validation

}

void loop() {
  Serial.println("\nStarting connection to server...");
  // If authentication is required
  // if (mqtt_client.connect("ESP32", mqtt_user , mqtt_pass)) {
  if (mqtt_client.connect("ESP32")) {
    Serial.print("Connected, mqtt_client state: ");
    Serial.println(mqtt_client.state());
    // Publish a test message to topic LivingRoom/TEMPERATURE
    mqtt_client.publish("test/topic", "25");
  }
  else {
    Serial.println("Connection failed! MQTT client state:");
    Serial.print(mqtt_client.state());
    Serial.println("\nWiFiClientSecure client state:");
    char lastError[100];
    client.lastError(lastError, 100);  // Get the last error for WiFiClientSecure
    Serial.print(lastError);
  }
  delay(10000);
}
```