

# Lab 3 – Connect with PUF protected PEM

## Import the WiFi with PUF project

- 1.1. Open a new Instance of MCUXpresso and Make a new Workspace “LPCLab3”
- 1.2. Use the Quickstart Panel to **Import projects from file system...**
  - 1.2.1. Navigate to \Desktop\LPC55Sxx E2E Hand s-on \PART3\lpcxpresso55s69\_aws\_remote\_control\_wifi\_PUF
  - 1.2.2. Select Next then **Finish** to import the SDK example project

## Update public credentials

Three (3) items must be updated.

Thing Name, REST API endpoint (Broker Endpoint) and The Device Certificate.

Thing Name and REST API endpoint are in `aws_client_credential.h`.

### Set these to match your Thing Name and Endpoint

```
35 /*
36  * MQTT Broker endpoint.
37  */
38 static const char clientcredentialMQTT_BROKER_ENDPOINT[] = "a3w3un1x9x1150-ats.iot.us-east-1.amazonaws.com";
39
40
41 /* Use of a "define" and not a "static const" here to be able to
42  * use pre-compile concatenation on the string. */
43 #define clientcredentialIOT_THING_NAME "Donnie"
..
```

In `aws_client_credential_keys.h`, update the Client Certificate PEM File information.

Copy the info from the previous AWS project that we used in LAB 1.

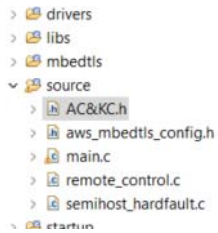
```
static const char clientcredentialCLIENT_CERTIFICATE_PEM[] =
"-----BEGIN CERTIFICATE-----\n"
"MIIDWjCCAAKqAwIBAgIWAOCMFAtcnEzqzh3YVAQ0VNRr1TsgMA0GCSqGSIb3DQEB\n"
"CwUAM0ExS2BjBGNVBAAsMQkFtYXpvbiBXX2MlglU2Vydm1jZXNmgTz1BbWf6b24uY29t\n"
"TElUy4gTD1TZWF0dGx1TFNUPVdhc2hpbmd0b24gQz1VUzAeFw0xOTA1MjgyMDIz\n"
"MzFw000TEyMzEyMzU5NT1aMB4xHDAaBgNVBAMME0FXUyBj1QgO2VydG1maWh\n"
"dGUvggE1MA0GCSqGSIb3DQEBAAQAA4IBDwAwggEKAoIBAQDZ+VLgVkyK0etK3Bw\n"
"N031h9c/gH2y+vMNg1XYH7esQf60Y1ke/veVPsFG6tjz++g9RQoghVp3I+PZg0\n"
"U4wrlwM/H2iMviiQZAFQvFHV18KD4SeBR6s2qcZ55KmkB9yj3gT4QXux3wVtdqAX\n"
"7+c1iVMeg80vbi3DFFPfyJEFAHI7DHYG7k9bo/1jkYCLMnZBfQnkuGGBsX+0qCL\n"
"1rHw/04Dss28nLq0aVpMOZFVXMA0+HmpETU1jtE0W5LC2MPTdh6bZfiYKq2+an8\n"
"1bu1vu048z9KLdPr15hmXiZTQ5ACqmrFD+asGGzfcYpNHM4k8CgHgKabQ1CHf+\n"
"hhA1AgMBAAGjYDBeMB8GA1UdIwQYMBaAF07i01mHvHACKrUJi6wzeqQLun0MB0G\n"
"A1UddgQB8B77PR/zFHMuOH9DASHNypffeUuQDAMBgNVHRMBAF8EAJAAMA4GA1Ud\n"
"DwEB/wQEAwIHgDANBgkqhkiG9w0BAQsFAAOCAQEATJwsMa8v0hHkQhPb19GvEa01\n"
"zlpJ3vK9Q1mKgj3yFawsgseS02pVcK1Zzv+u0E22/g95dnjHjh2cr5whjNEBYw0d\n"
"ChMBg1F07k1y1bx1UuU5m+z9+ApB8QAYmiOZ0LYi3APPR1guFHSjN5i7JLe1da+x\n"
"qr0HavN57gQ10kmB69Zi/8azy3e0Ut4KXXRG2AgSPCQunVEmeeRpK/ogWwC4WV\n"
"tsnqVnGkNIInw08pyNcz/MaFrHjbr0hJ8S4o9uIrfJkzJi8yBa0B/TWiM/iq2MC\n"
"6x76XUXhny6Ut13H/GT5YFqu4Twdm/cXGsz29VF/2a8Xqh8J2WoL4VVGyUjIA==\n"
"-----END CERTIFICATE-----";

/*
 * PEM-encoded client private key.
 *
 * Must include the PEM header and footer:
 * "-----BEGIN RSA PRIVATE KEY-----"
 * "...base64 data..."
 * "-----END RSA PRIVATE KEY-----";
 */
static const char clientcredentialCLIENT_PRIVATE_KEY_PEM[] =
"";
```

## Update the PUF Data

Open the C file data created by Hex Editor (you can open from MCUXpresso, File-> Open File

Copy Array data and paste inside AC&KC.h file that is in the source folder



```
4
5 __RODATA(Flash2) static const uint8_t AC_DATA[4096] = {
6
7     0x3F, 0x41, 0xC8, 0xB2, 0x16, 0x17, 0xD8, 0x90, 0x27, 0xFA, 0x3F, 0x7F,
8     0x19, 0x2B, 0x8C, 0xE2, 0x0B, 0x83, 0xF5, 0xA9, 0x52, 0x0D, 0x6F, 0x84,
9     0xC1, 0x98, 0x7C, 0x8F, 0x75, 0xF7, 0x76, 0xA6, 0x9C, 0x45, 0x0A, 0x80,
0     0xFB, 0xCD, 0xD7, 0x5D, 0x2F, 0xB5, 0xC2, 0x02, 0x4F, 0x97, 0x2D, 0xBF,
1     0x2F, 0x40, 0x5B, 0x1B, 0x84, 0x06, 0xD1, 0x18, 0xA8, 0x87, 0x45, 0x4B,
2     0x79, 0x27, 0x6E, 0x63, 0x51, 0x06, 0xDD, 0xBC, 0x95, 0xBE, 0xFB, 0x96,
3     0x5D, 0x91, 0x6F, 0x46, 0xF7, 0x81, 0xA4, 0x17, 0xA0, 0x57, 0x9C, 0xEB,
4     0x89, 0x9A, 0x7B, 0xFC, 0xED, 0xDB, 0xC6, 0xA8, 0x26, 0xB1, 0xCE, 0x43,
5     0x80, 0xE7, 0x2A, 0x06, 0x19, 0x16, 0xA7, 0xDB, 0x58, 0xBF, 0x09, 0x79,
6     0xDB, 0xF2, 0x27, 0xFF, 0x2E, 0x06, 0x35, 0x33, 0x62, 0x79, 0x5C, 0x86,
7     0xF7, 0x50, 0x9B, 0x9E, 0xDE, 0x33, 0xD5, 0xF2, 0xE2, 0x10, 0xC5, 0x4C,
8     0xE9, 0x99, 0x6E, 0xFC, 0xD9, 0x69, 0x2D, 0xF1, 0x78, 0xA3, 0x92, 0x3D,
9     0x8B, 0xA3, 0x06, 0x37, 0x82, 0x1F, 0xDA, 0x0C, 0x9C, 0xFF, 0xDA, 0x0B,
0     0xA1, 0xB7, 0x10, 0xD3, 0x9C, 0x96, 0xF7, 0x40, 0xC3, 0xD9, 0xF2, 0x6B,
1     0x39, 0x67, 0x0A, 0x71, 0x3E, 0x16, 0x1B, 0x8A, 0x96, 0xAE, 0x1D, 0x80,
2     0xA6, 0xE0, 0x7B, 0xE0, 0x3D, 0x71, 0x28, 0x58, 0x0C, 0xA3, 0x76, 0xD3,
3     0x0B, 0x83, 0x80, 0x33, 0xAF, 0x07, 0x8D, 0x6B, 0x4B, 0x4D, 0x53, 0x53
```

## Download and Debug with PUF info only



With the project highlighted - Press the Blue Bug and run the project.

```
Key:
0: 6d 74 78 2f 49 6d 38 4e 2b 55 43 76 55 64 6d 42
16: 51 41 66 67 6d 46 33 6b 4a 51 61 33 70 63 4c 4a
32: 41 5a 71 37 32 63 58 52 75 42 74 33 4b 38 48 4d
48: 6c 41 5a 4a 0 20 20 20 0 0 [Tmr Svc] Starting key provisioning...
1 0 [Tmr Svc] Write root certificate...
2 10 [Tmr Svc] Write device private key...
3 424 [Tmr Svc] Write device certificate...
4 435 [Tmr Svc] Key provisioning done...
5 436 [Tmr Svc] Starting WiFi...
6 1705 [Tmr Svc] WiFi module initialized.
7 6734 [Tmr Svc] WiFi connected to AP nxp.
8 6735 [Tmr Svc] IP Address acquired 192.168.43.120
9 6744 [AWS-RemoteCtrl] [Shadow 0] MQTT: Creation of dedicated MQTT client succeeded.
10 7133 [MQTT] Looked up a3w3unlx9x1150-ats.iot.us-east-1.amazonaws.com as 54.242.222.183
11 23959 [AWS-RemoteCtrl] [Shadow 0] MQTT: Connect succeeded.
12 24169 [AWS-RemoteCtrl] [Shadow 0] MQTT: Subscribe to accepted topic succeeded
13 24381 [AWS-RemoteCtrl] [Shadow 0] MQTT: Subscribe to rejected topic succeeded
14 24390 [AWS-RemoteCtrl] [Shadow 0] MQTT: Publish to operation topic succeeded.
```