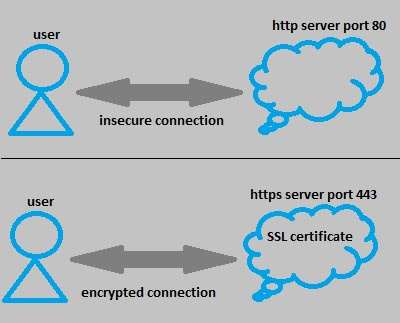
# **[Demo 29: How to use HTTPS in Arduino ESP32](http://www.iotsharing.com/2017/08/how-to-use-https-in-arduino-esp32.html)**

**1. Introduction**  
In this tutorial I will show you how to use HTTPS for secure communication with server.

[](https://1.bp.blogspot.com/-LMDzw8xOxs8/WZzd5OrdGLI/AAAAAAAAENo/3sx-PVeRu7s-8-5bMVRgoguqHx8UELTLQCLcBGAs/s1600/esp32-https_1.jpg)

**Figure: HTTP vs HTTPS**

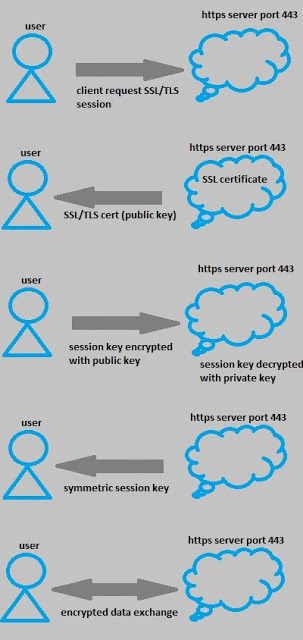
**How it work?**  
In order to understand more clear, you should read about [SSL&TLS](http://blogs.mdaemon.com/index.php/ssl-tls-best-practices/" \t "http://www.iotsharing.com/2017/08/_blank) and [HTTPS](https://en.wikipedia.org/wiki/HTTPS" \t "http://www.iotsharing.com/2017/08/_blank). Here are some main points:

- HTTPS refers to use of ordinary HTTP over an encrypted SSL (Secure Sockets Layer) or TLS (Transport Layer Security) connection.  
- TLS was introduced in 1999 as a new version of SSL and was based on SSL 3.0.    
- A concept called **SSL/TLS certificate** which is used to establish a SSL/TLS connection. SSL/TLS certificates use a key pair (a public and private key) to encrypt/decrypt data before exchanging it. So certificates are not dependent on protocols. It means when replacing SSL by TLS, the certificate is not change.

[](https://3.bp.blogspot.com/-0LLW2zKWTzo/WZ37uYzqMlI/AAAAAAAAEOI/sKcCt9qRXpcR5FJ3GM74R8qLGPhyzZthgCLcBGAs/s1600/cert.png)

**Figure: SSL/TLS certificate content**

- The HTTPS message is encrypted, including the headers, and the request/response payload.

[](https://2.bp.blogspot.com/-sW7V-EwmpVk/WZ0pI_wn3VI/AAAAAAAAEN4/m6-E79nxwTQ_C219lnQTe-fMv4L5brIaACLcBGAs/s1600/esp32-https_2.jpg)

**Figure: work-flow of HTTPS**

- The process before exchanging data is called **SSL handshake**

- SSL/TLS Certificates need to be issued by a trusted Certificate Authority (CA). Client devices (browsers, OS, ... ) maintains list of trusted CA root certificates so that they can compare with server certificates in SSL handshake phase.

- We also have [Self-Signed SSL Certificate](https://en.wikipedia.org/wiki/Self-signed_certificate) which is created by own self. It is considered insecure.It can be generated using openssl tool.

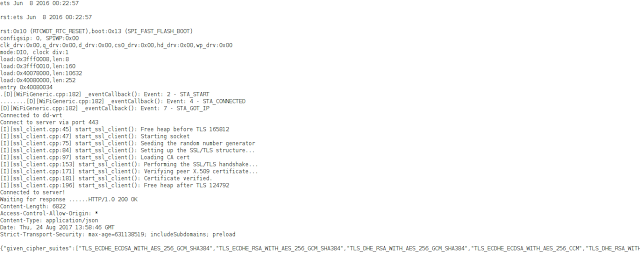
- In order to get certificate of a website that we want to establish a HTTPS connection. We will use **openssl** tool.  
- You can install this tool for:  
**Ubuntu: sudo apt-get install openssl**  
**CenOS ReadHat: yum install openssl**  
**Windows: you can refere [here](http://www.iotsharing.com/2017/06/4-steps-to-install-mosquitto-mqtt-on-windows.html)**  
- In this demo we will create a simple HTTPS request from ESP32 client to "[https://www.howsmyssl.com](https://www.howsmyssl.com/)[/a/check"](https://www.blogger.com/null) (this site is used to check HTTPS connection) and print the response to Terminal.    
**2. Software**  
- ESP32 using mbedTLS for SSL handshake phase. It is wrapped under **WiFiClientSecure**class.  
- We create an instance of WiFiClientSecure:  
**WiFiClientSecure client;**  
and then we call method: **client.setCACert(content\_of\_certificate)**to point to SSL/TLS certificate for SSL handshake phase. Then we call a set of functions to form a HTTPS request:  
**client.println("GET https://www.howsmyssl.com/a/check HTTP/1.0");**  
**client.println("Host: www.howsmyssl.com");**  
**client.println("Connection: close");**  
Finally, we get the response from server using:  
**client.available()**: to check whether server response data or not.  
**client.read()**: to read response data from server**.**  
In order to get the SSL/TLS certificate of "[https://www.howsmyssl.com](https://www.howsmyssl.com/)[/a/check" we will use openssl tool above.](https://www.blogger.com/null)From command line typing: **openssl s\_client -showcerts -connect www.howsmyssl.com:443**

[](https://4.bp.blogspot.com/-Vkzeouo2Rlw/WZ7cPsq5I2I/AAAAAAAAEOc/KR1E9-AHI1Y8A_BJbcAJgzfrPr_kB_i8ACLcBGAs/s1600/esp32-https-4.png)

**Figure: Choose the certificate that contains the Signature Trust line**

|  |
| --- |
| #include <WiFiClientSecure.h>  const char\* ssid = "your-ssid"; // your network SSID  const char\* password = "your-password"; // your network password  const char\* server = "www.howsmyssl.com"; // Server URL  /\* use  openssl s\_client -showcerts -connect www.howsmyssl.com:443 </dev/null  to get this certificate \*/  const char\* ca\_cert = \  "-----BEGIN CERTIFICATE-----\n" \  "MIIEkjCCA3qgAwIBAgIQCgFBQgAAAVOFc2oLheynCDANBgkqhkiG9w0BAQsFADA/\n" \  "MSQwIgYDVQQKExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNVBAMT\n" \  "DkRTVCBSb290IENBIFgzMB4XDTE2MDMxNzE2NDA0NloXDTIxMDMxNzE2NDA0Nlow\n" \  "SjELMAkGA1UEBhMCVVMxFjAUBgNVBAoTDUxldCdzIEVuY3J5cHQxIzAhBgNVBAMT\n" \  "GkxldCdzIEVuY3J5cHQgQXV0aG9yaXR5IFgzMIIBIjANBgkqhkiG9w0BAQEFAAOC\n" \  "AQ8AMIIBCgKCAQEAnNMM8FrlLke3cl03g7NoYzDq1zUmGSXhvb418XCSL7e4S0EF\n" \  "q6meNQhY7LEqxGiHC6PjdeTm86dicbp5gWAf15Gan/PQeGdxyGkOlZHP/uaZ6WA8\n" \  "SMx+yk13EiSdRxta67nsHjcAHJyse6cF6s5K671B5TaYucv9bTyWaN8jKkKQDIZ0\n" \  "Z8h/pZq4UmEUEz9l6YKHy9v6Dlb2honzhT+Xhq+w3Brvaw2VFn3EK6BlspkENnWA\n" \  "a6xK8xuQSXgvopZPKiAlKQTGdMDQMc2PMTiVFrqoM7hD8bEfwzB/onkxEz0tNvjj\n" \  "/PIzark5McWvxI0NHWQWM6r6hCm21AvA2H3DkwIDAQABo4IBfTCCAXkwEgYDVR0T\n" \  "AQH/BAgwBgEB/wIBADAOBgNVHQ8BAf8EBAMCAYYwfwYIKwYBBQUHAQEEczBxMDIG\n" \  "CCsGAQUFBzABhiZodHRwOi8vaXNyZy50cnVzdGlkLm9jc3AuaWRlbnRydXN0LmNv\n" \  "bTA7BggrBgEFBQcwAoYvaHR0cDovL2FwcHMuaWRlbnRydXN0LmNvbS9yb290cy9k\n" \  "c3Ryb290Y2F4My5wN2MwHwYDVR0jBBgwFoAUxKexpHsscfrb4UuQdf/EFWCFiRAw\n" \  "VAYDVR0gBE0wSzAIBgZngQwBAgEwPwYLKwYBBAGC3xMBAQEwMDAuBggrBgEFBQcC\n" \  "ARYiaHR0cDovL2Nwcy5yb290LXgxLmxldHNlbmNyeXB0Lm9yZzA8BgNVHR8ENTAz\n" \  "MDGgL6AthitodHRwOi8vY3JsLmlkZW50cnVzdC5jb20vRFNUUk9PVENBWDNDUkwu\n" \  "Y3JsMB0GA1UdDgQWBBSoSmpjBH3duubRObemRWXv86jsoTANBgkqhkiG9w0BAQsF\n" \  "AAOCAQEA3TPXEfNjWDjdGBX7CVW+dla5cEilaUcne8IkCJLxWh9KEik3JHRRHGJo\n" \  "uM2VcGfl96S8TihRzZvoroed6ti6WqEBmtzw3Wodatg+VyOeph4EYpr/1wXKtx8/\n" \  "wApIvJSwtmVi4MFU5aMqrSDE6ea73Mj2tcMyo5jMd6jmeWUHK8so/joWUoHOUgwu\n" \  "X4Po1QYz+3dszkDqMp4fklxBwXRsW10KXzPMTZ+sOPAveyxindmjkW8lGy+QsRlG\n" \  "PfZ+G6Z6h7mjem0Y+iWlkYcV4PIWL1iwBi8saCbGS5jN2p8M+X+Q7UNKEkROb3N6\n" \  "KOqkqm57TH2H3eDJAkSnh6/DNFu0Qg==\n" \  "-----END CERTIFICATE-----\n";  /\* create an instance of WiFiClientSecure \*/  WiFiClientSecure client;  void setup() {  Serial.begin(115200);  WiFi.begin(ssid, password);  /\* waiting for WiFi connect \*/  while (WiFi.status() != WL\_CONNECTED) {  Serial.print(".");  delay(100);  }  Serial.print("Connected to ");  Serial.println(ssid);  /\* set SSL/TLS certificate \*/  client.setCACert(ca\_cert);  Serial.println("Connect to server via port 443");  if (!client.connect(server, 443)){  Serial.println("Connection failed!");  } else {  Serial.println("Connected to server!");  /\* create HTTP request \*/  client.println("GET https://www.howsmyssl.com/a/check HTTP/1.0");  client.println("Host: www.howsmyssl.com");  client.println("Connection: close");  client.println();  Serial.print("Waiting for response ");  while (!client.available()){  delay(50); //  Serial.print(".");  }  /\* if data is available then receive and print to Terminal \*/  while (client.available()) {  char c = client.read();  Serial.write(c);  }  /\* if the server disconnected, stop the client \*/  if (!client.connected()) {  Serial.println();  Serial.println("Server disconnected");  client.stop();  }  }  }  void loop() {  } |

**3. Result**

[](https://1.bp.blogspot.com/-boc0zVqUlHs/WZ7cQpUPh7I/AAAAAAAAEOg/DPq3-VuGW5ky7hMp182y5Psmp6M9-HvIACEwYBhgL/s1600/esp32-https-3.png)

**Figure: Request and response from server**