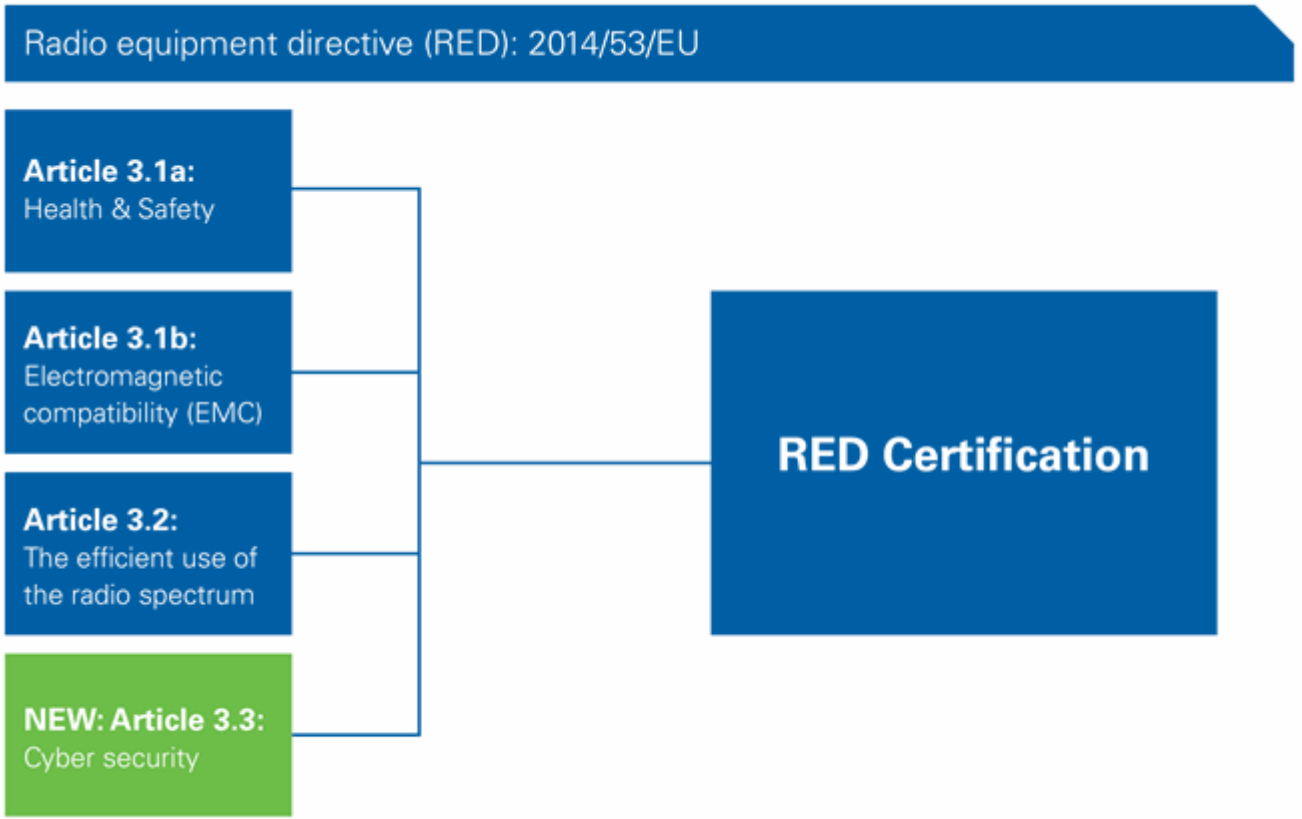
Un semplice client TLS su ESP32

Attività progettuale di Cyber Security

Andrea Zanni

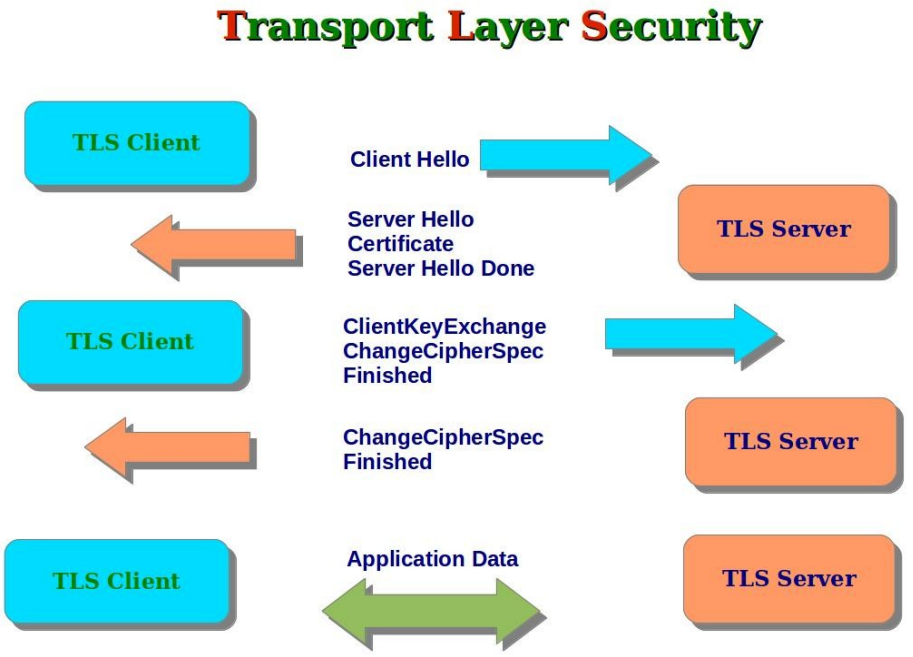
a.a. 2022/2023

RED – Radio Equipment Directive

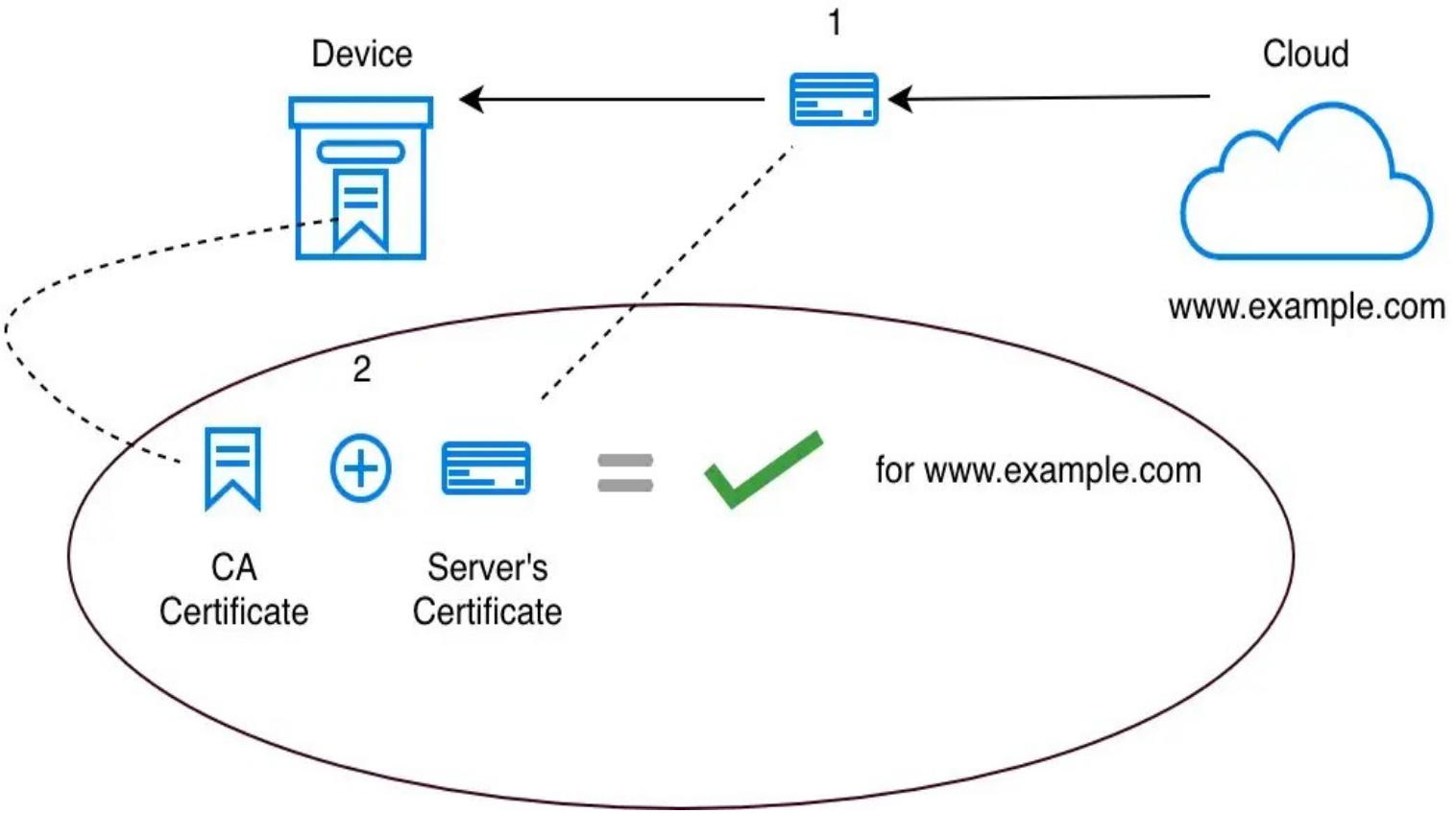


https://www.kiwa.com/nl/en/themes/cyber-security/news/red-delegated-act-mandatory-c
to-articles-3-3-d-e-and-f-inbound/

TLS


Transport Layer Security
TLS Client
Client Hello
TLS Server
Server Hello
Certificate
Server Hello Done
TLS Client
ClientKeyExchange
ChangeCipherSpec
Finished
TLS Server
ChangeCipherSpec
Finished
TLS Client
Application Data
TLS Server

Certificati (1)



Source https://blog.espressif.com/esp32-tls-transport-layer-security-and-iot-devices-3ac9

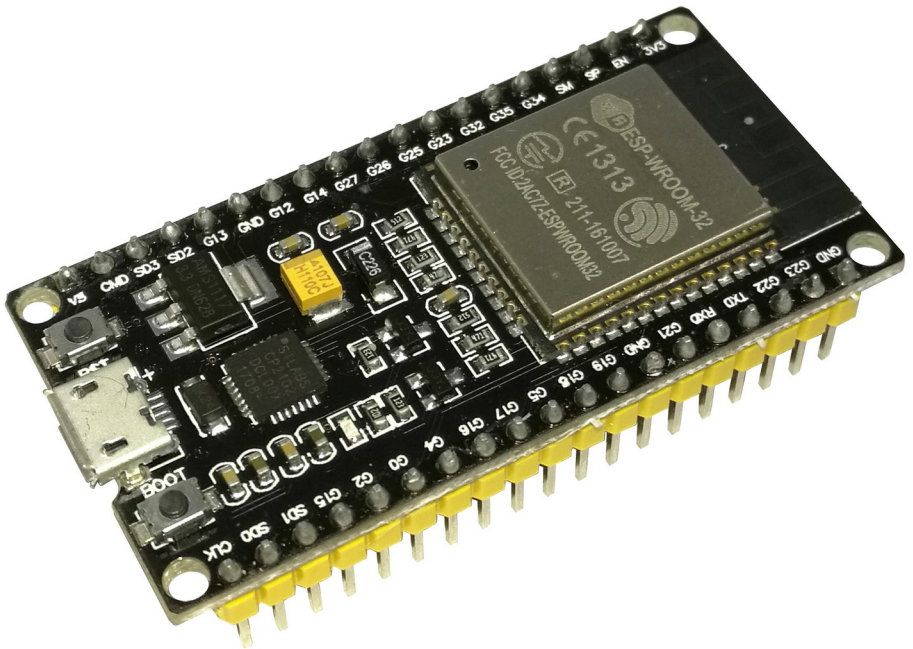
Certificati (2)

- Creata la CA si procede con la creazione del certificato auto-firmato per il dominio sslserver.home

```
$ openssl x509 -req -sha512 -days 50 -in host.csr -CA ../../main/ca.crt -CAkey ../../main/ca.key -CAcreateserial -out host.crt -
extfile host-ext.conf
```

```
x509: certificato di tipo x509

-days: scadenza (in giorni)

-in: la richiesta di firma del certificato (CSR: Certificate Signing Request)

-CA: il certificato della Certifying Authority (noi)

-Cakey: la chiave della CA per firmare il certificato

-extfile: file di configurazione aggiuntivo necessario per creare un certificato firmato per il dominio sslserver.home

-out: produce il certificato per il server
```

La board di sviluppo - ESP32-WROOM-32



- 4 MiB di memoria flash
- 520 KiB di SRAM per dati e istruzioni
- 8 KiB di SRAM chiamata RTC FAST Memory
- 8KiB di SRAM chiamata RTC SLOW Memory
- ULP co-processor

Embedding dati binari

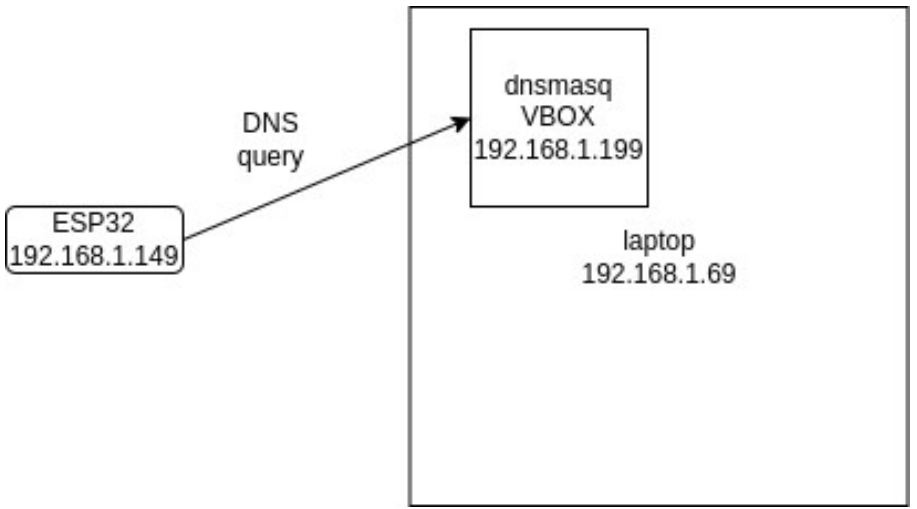
“If your CA certificate is embedded in the device firmware, you can update the CA certificates by doing an OTA firmware upgrade. The new firmware can contain the updated CA certificate to be used.”

- Per ottenere il vantaggio di aggiornare I certificati delle CA Over-The-Air ho incastonato il certificato nel firmware poi flashato sulla board

```
target_add_binary_data(esp-tls-test.elf "main/ca.crt" TEXT)
```

- Viene prodotto il file ca.crt.S, al suo interno notiamo le sezioni richiamate dal codice. Notiamo anche che il file binario fa parte della sezione .rodata.embedded
- NOTA: esp-tls-test.elf non sarà il firwmare flashato sulla board ma solo una sua parte.

DNS (1)



- L’API ESP-TLS utilizza mbedTLS come implementazione del protocollo TLS. A sua volta, mbedTLS utilizza lwip per risolvere il nome DNS il quale si riduce a una getaddrinfo().

DNS (2) – Traffico di rete

eth.src == ec:62:60:84:dd:ac || eth.dst == ec:62:60:84:dd:ac

No.	Time	Source	Destination	Protocol	Length	Info
833	9.843657721	Espressi_84:dd:ac	Broadcast	ARP	42	Who has 192.168.1.149? (ARP Probe)
851	10.245303749	Espressi_84:dd:ac	Broadcast	ARP	42	Who has 192.168.1.149? (ARP Probe)
857	10.662847567	Espressi_84:dd:ac	Broadcast	ARP	42	Who has 192.168.1.149? (ARP Probe)
861	10.753011185	Espressi_84:dd:ac	Broadcast	ARP	42	ARP Announcement for 192.168.1.149
948	10.860381295	Espressi_84:dd:ac	Broadcast	ARP	42	Who has 192.168.1.199? Tell 192.168.1.149
970	10.861070786	IntelCor_8f:b5:7e	Espressi_84:dd:ac	ARP	60	192.168.1.199 is at ac:74:b1:8f:b5:7e
1004	10.962172248	192.168.1.149	192.168.1.199	DNS	74	Standard query 0xa5fb A sslserver.home
1006	10.963030915	192.168.1.199	192.168.1.149	DNS	90	Standard query response 0xa5fb A sslserver.home A 192.168.1.69
1007	10.963102921	192.168.1.199	192.168.1.149	DNS	90	Standard query response 0xa5fb A sslserver.home A 192.168.1.69
1008	10.966989028	192.168.1.149	192.168.1.199	ICMP	70	Destination unreachable (Port unreachable)
1010	11.064428401	Espressi_84:dd:ac	Broadcast	ARP	42	Who has 192.168.1.69? Tell 192.168.1.149
1011	11.064469778	IntelCor_8f:b5:7e	Espressi_84:dd:ac	ARP	42	192.168.1.69 is at ac:74:b1:8f:b5:7e
1013	11.167637817	192.168.1.149	192.168.1.69	TCP	58	63722 → 6000 [SYN] Seq=0 Win=5744 Len=0 MSS=1440
1014	11.167727816	192.168.1.69	192.168.1.149	TCP	54	6000 → 63722 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1019	11.676014655	Espressi_84:dd:ac	Broadcast	ARP	42	ARP Announcement for 192.168.1.149
1188	13.726883826	Espressi_84:dd:ac	Broadcast	ARP	42	ARP Announcement for 192.168.1.149
1567	15.971920441	IntelCor_8f:b5:7e	Espressi_84:dd:ac	ARP	60	Who has 192.168.1.149? Tell 192.168.1.199
1578	16.079953738	Espressi_84:dd:ac	IntelCor_8f:b5:7e	ARP	42	192.168.1.149 is at ec:62:60:84:dd:ac
7746	63.899643801	Espressi_84:dd:ac	Broadcast	ARP	42	ARP Announcement for 192.168.1.149

Server

```
$ openssl s_client -CAfile certs/ca.crt -connect
0.0.0.0:6000

s_client: implementazione di un client TLS

-Cafile: specificare il certificato della CA da usare per
validare il certificato del server

-connect: connessione al server in locale specificato
successivamente
```

- Comando utilizzato per verificare il corretto funzionamento del server.
- Server implementato in python3
- https://github.com/andreaz98/esp32-tls/blob/master/tls-server/sslserver2.py

Client (1)

- Implementato nel caro e vecchio C
- ESP-TLS API
- Lwip API

- https://github.com/andreaz98/esp32-tls/blob/master/main/esp-tls-test.c

Client (2) – Verifica del certificato

```
I (7094) esp_netif_handlers: sta ip: 192.168.1.149, mask: 255.255.255.0, gw: 192.168.1.1
[WIFI] Connected to WiFi
W (7414) wifi:<ba-add>idx:0 (ifx:0, 10:71:b3:d4:4c:e1), tid:0, ssn:5, winSize:64
E (23074) esp-tls-mbedtls: mbedtls_ssl_handshake returned -0x0050
I (23084) esp-tls-mbedtls: Certificate verified.
E (23084) esp-tls: Failed to open new connection
E (23084) esp-tls-mbedtls: write error :-0x0050:
```

```
esp_tls_cfg_t cfg = {
        .use_global_ca_store = true
};
```

- Use_global_ca_store: Flag da settare per verificare il certificato del server con la CA precedentemente embeddata
- Vi è un opportuno puntatore a cui far puntare il buffer contenente il certificato del client così da realizzare la mutua autenticazione

Estensioni future

- Autenticazione del client agendo su esp_tls_cfg_t
- Creazione di una chain di certificati di CA per poter verificare server autenticati da CA diversa dalla nostra oltre al x509 cert bundle
- Rendere la chiamata asincrona per un client più performante

- Codice sorgente del client e del server, certificati, file di configurazione sono in questa repository https://github.com/andreaz98/esp32-tls

DEMO :)

Dimostrazione :)