IRELE

Mobile and wireless network

Project report

Authors:

Arico Amaury

Mototani Yuta

Tchakounte Loic

Sipakam Cedric

Professor:

JM. Dricot

TA:

Ladner Navid

Academic year:

2024 - 2025

Contents

1 Introduction and concept

1.1 Device

1.2 Concept

2 Threat Model

2.1 Data flow diagram

In the first chapter, we discussed about the project concept where one ESP32 is used as a data publisher mimicking the data collection of a room we would like the thermal conditions to be controlled via a command center, which is the alias for our second ESP32.
The acquised data are sent from our ESP32 "publisher" to a MQTT server, installed on our machine for the example. The second ESP32 is subscribed to the MQTT server, receiving the published data and in charge of monitoring - controlling the room conditions. Those data are displayed on another MQTT server able to plot data evolution.
The data flow is shown on the figure 2.1

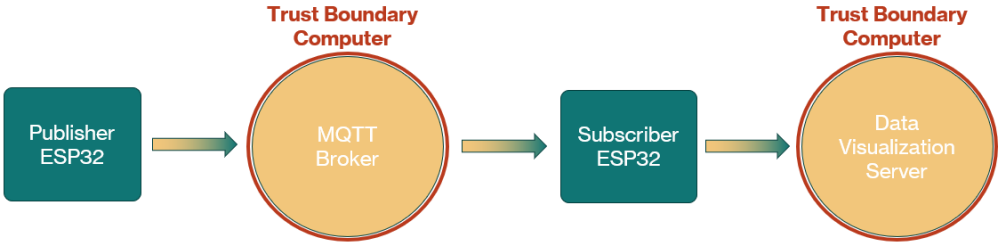

Figure 2.1: Data flow diagram

For completeness, we will consider our data as highly sensitive - with strict security needs - where the room thermal regulation cannot be vulnerable to seizing control attacks or unstablizing temperature attacks for example.
We would therefore need to mitigate all types of attack by integrating a security strategy considering the ressources constraints dependent on the hardware used - in our case, the ESP32 devkit with :

1. 4MB flash memory
2. 520KB RAM
3. Dual-core 32-bit processor

Thus HTTPS / TLS protocol are not adapted for our devices and key cryptoghraphy containing complex mathematical operation like RSA cannot be implemented. This suitable security strategy counter acting the whole span of threats while fitting into the hardware and meeting the decyphering process speed will be discussed in the last chapter namely the "Security mitigation".

Before treating about the security strategy, we would need to identify the security threats our model needs to deal with. The STRIDE model will be used in this perspective in the next section.

Figure 2.2

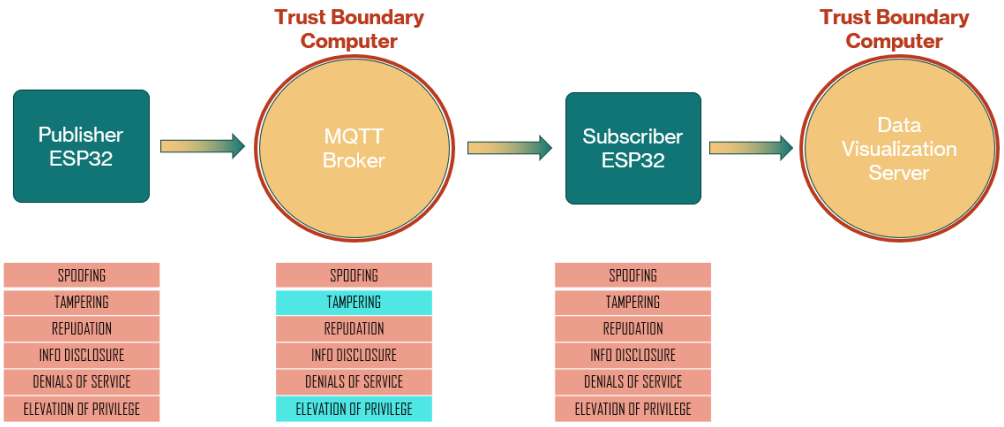

Figure 2.2: STRIDE model verification for project data flow

2.2 STRIDE Threat Model

3 Security Mitigation

3.1 Security scheme

Figure 3.1

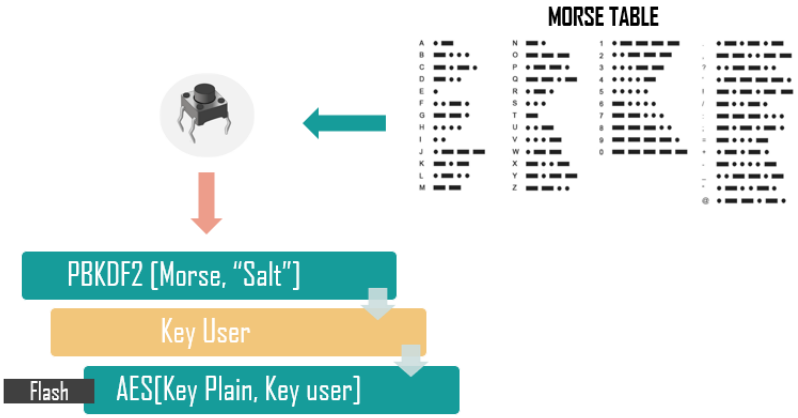

Figure 3.1: Encrypted key generation

Figure 3.2

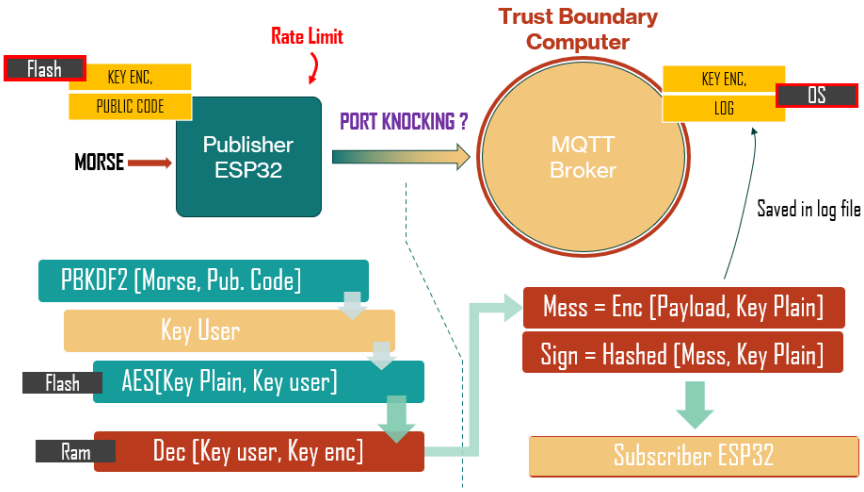

Figure 3.2: Security strategy integration in the project data flow

3.2 Results

Figure 3.3

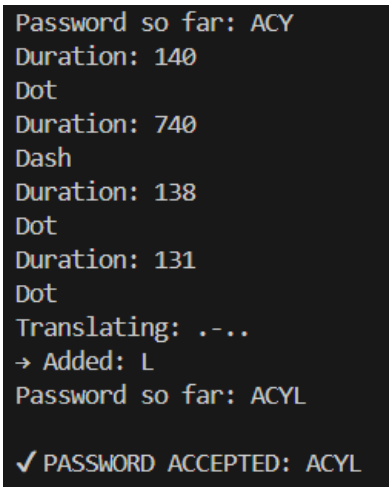

Figure 3.3: Morse code implementation Result

Figure 3.4

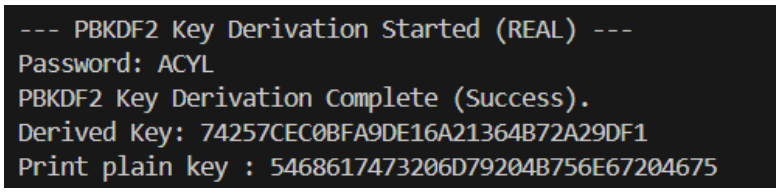

Figure 3.4: PBKDRF2 encryption verification

Figure 3.5

Figure 3.5: Encrypted data publishment on MQTT server

Figure 3.6

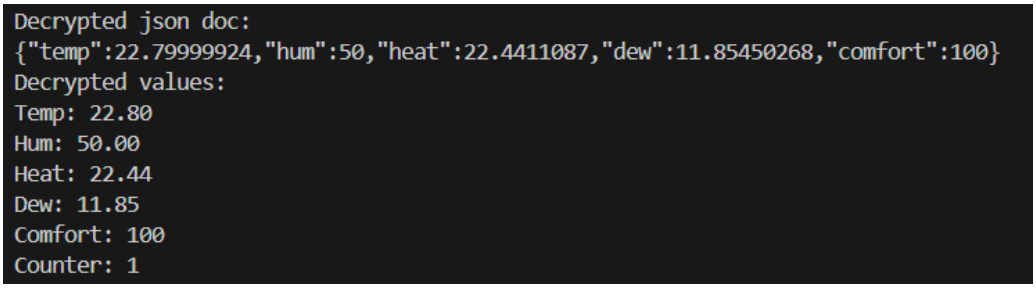

Figure 3.6: Decryption verification

Figure 3.7

```
uint32_t counter_rx;
memcpy(&counter_rx, decoded + 16, sizeof(uint32_t));

uint32_t rx_last_counter = rx_glob_counter;

if (counter_rx <= rx_glob_counter) {
    Serial.println("Replay detected");
    free(decoded);
    return String("");
} else {
    rx_glob_counter = counter_rx;
    write_counter_flash_rx(rx_glob_counter);
}

if (out_counter_rx != nullptr){
    *out_counter_rx = counter_rx;
}
```

Figure 3.7: Counter implementation for replay counter strategy

Figure 3.8

```
Decrypted json doc:
{"temp":22.79999924,"hum":50,"heat":22.4411087,"dew":11.85450268,"comfort":100}
Decrypted values:
Temp: 22.80
Hum: 50.00
Heat: 22.44
Dew: 11.85
Comfort: 100
Counter: 1
Switching Suscribed port : 1900
Connecting to MQTT...Connected!
 Temp:22.50 Hum:52.00 Index:22.16 Dew:12.17 Comfort_OK
 Temp:22.60 Hum:50.00 Index:22.22 Dew:11.67 Comfort_OK
***** PUBLISHMENT OF PLAIN DATA TO MQTT SERVER****
Sending data to Port 1950 for data Visualisation
```

Figure 3.8: Port switching for sniffing security improvement

3.3 Discussion

3.4 Conclusion