Security and Encryption

Introduction

Security is an important aspect of an IoT communication platform. After all, if a network is not properly secured, any attacker can sniff data from the sensors, or send potentially disruptive commands to the actuators, causing damage to the system.

The security of any SPRN network revolves around a pre-shared, 240-bit network secret that is stored in every device once it is configured via UART. This secret is then used to generate a packet key for every sent packet via SHA-256. Due to the hardware implementation of the AES cipher and SHA algorithms in ESP32 devices, and the relatively low throughput of the radio network, this encryption scheme does not create a bottleneck for packet transmission.

Security Features

Considering that IoT devices are predicted to become ubiquitous in industry, farming, services, and our everyday lives, it is required of a network to provide sufficient security against possible attackers, so that its sensor data cannot be eavesdropped, its actuators cannot be manipulated to work in an unwanted manner, and its transmitted data cannot be changed by a third-party at will.

Security against eavesdroppers

Considering that this network is using a widely-used unlicensed radio network, it is expected that any individual or organization would have access to devices that can read the transmitted packets. However, The only non-encrypted data that is transmitted in an SPRN network is the length of each packet, the network ID, and the sequence number of a packet, giving an eavesdropper only minimal data, i.e. the network ID, which might as well be publicly known, the packet length, which the eavesdropper can manually measure, and the sequence number, which is of no use to any eavesdropper. Considering that not only the contents, but the nature of the packets are unknown to the eavesdropper, and the fact that the sequence number is used to generate a new key for every transmission, the only distinction that an eavesdropper can make between different packages is the distinction between those with a payload (MSG & CMD) and those without one.

Security Against External Packets

As said before, the only part of a packet that a third-party can access without having the secret key, consists of publicly known data. To generate a fake packet from only the known parts of a packet, an adversary would have to brute force their way through an AES-256 cipher, and considering the key generation scheme, an adversary would have a maximum of 15 blocks of data for doing so, making it an unfeasible task.

Another type of attack (replay attacks) might be envisioned against this network. However, in doing so, an adversary would have to wait a certain period of time, until the sequence number has ticked about 16 million times. And even then, the timestamp in the old package would make it obvious that the sent packet has expired.

Security Against Man-in-the-Middle Attacks

While external devices cannot send or receive data without authorization, there is a possibility where a device would enter as a middleman, and direct or block packets, causing unexpected behavior in the network, blocking available addresses, and blocking resources in the parent nodes. However, considering that a device would have to encrypt parts of the SRCH and ADP packets, it is not possible for a third-party device to pose as a new device or a connected device in the network.

A third-party can however retransmit a device's SRCH packet and send the received ADP response from an out-of-range device, hoping that the new device would choose its ADP. This approach can be used to misdirect a network and cause temporary packet loss, however, since the ACK packets cannot be produced by a middleman, both the server and the device would become aware of the packet loss.

Key Generation and Cipher Operations

One of the basic principles in communicatoin security is to limit the amount of data encrypted with one key. Doing so would require the network to assign new keys periodically. The way this is done in an SPRN network, without the need for assymmetric encryption or synchronized networks, is by using a 240-bit preshared secret and per-packet sequence numbers. The 8-bit part in the beginning of each packet is concatenated to the end of the secret, and passed through a SHA-256 hash function. The output is then used as the AES-256 key for the encryption of a maximum of 240 bytes of data.

$$K = Hash(Secret || Salt)$$

This is of course a relatively costly operation. However, considering the limited throughput of the SPRN network, and the fact that ESP32 chips have hardware accelerators for cryptographic operations, doing so would not put a bottleneck on the network's performance.

The AES-256 encryption and decryption is done each time a device receives or sends a packet. The encryption is done in CBC mode, with an IV of zero bytes.