LoRaWAN Security

LoRaWAN 1.0 specifies a number of security keys: `NwkSKey`, `AppSKey` and `AppKey`. All keys have a length of 128 bits.

The **Network Session Key** (`NwkSKey`) is used for interaction between the Node and the Network Server. This key is used to check the validity of messages (MIC check). In the backend of The Things Network this validation is also used to map a non-unique device address (`DevAddr`) to a unique `DevEUI` and `AppEUI`.

The **Application Session Key** (`AppSKey`) is used for encryption and decryption of the payload. The payload is fully encrypted between the Node and the Handler/Application Server component of The Things Network (which you will be able to run on your own server). This means that nobody except you is able to read the contents of messages you send or receive.

These two session keys (`NwkSKey` and `AppSKey`) are unique per device, per session. If you dynamically activate your device (OTAA), these keys are re-generated on every activation. If you statically activate your device (ABP), these keys stay the same until you change them.

Dynamically activated devices (OTAA) use the **Application Key** (`AppKey`) to derive the two session keys during the activation procedure. In The Things Network you can have a *default* `AppKey` which will be used to activate all devices, or customize the `AppKey` per device.

Frame Counters

Because we're working with a radio protocol, anyone will be able to capture and store messages. It's not possible to read these messages without the `AppSKey` , because they're encrypted. Nor is it possible to tamper with them without the `NwkSKey` , because this will make the MIC check fail. It is however possible to re-transmit the messages. These so-called replay attacks can be detected and blocked using frame counters.

When a device is activated, these frame counters (`FCntUp` and `FCntDown`) are both set to `0`. Every time the device transmits an uplink message, the `FCntUp` is incremented and every time the network sends a downlink message, the `FCntDown` is incremented. If either the device or the network receives a message with a frame counter that is lower than the last one, the message is ignored.

This security measure has consequences for development devices, which often are statically activated (ABP). When you do this, you should realize that these frame counters reset to `0` every time the device restarts (when you flash the firmware or when you unplug it). As a result, The Things Network will block all messages from the device until the `FCntUp` becomes higher than the previous `FCntUp`. Therefore, you should re-register your device in the backend every time you reset it.

Spread Spectrum

Spread Spectrum Radio Transmission was traditionally used, during WW2, to make military communications difficult to monitor - either by using a technique called 'frequency hopping' (FHSS) - skipping the transmission frequency around in a prearranged manner, causing the enemy to constantly retune (very rapidly) or 'direct sequence' (DSSS) where the digital message is added to a much higher bit-rate, pseudo random (PR) sequence. The code spreads the radio signal over a much wider bandwidth. In fact, so wide that the power may well be dispersed so that the total signal falls down into the

background radio noise - and becomes invisible. Recovery is therefore a matter of i) knowing the original radio frequency ii) the pseudo random code and iii) and the PR code bit rate. Knowing these details means that synchronising receivers is not as difficult as may at first appear. The signal will just 'pop up' out of the noise when the correct values are achieved. ('Processing Gain')

The technique used in LoRa is 'CHIRP': Compressed High Intensity Radar Pulse. It is even more complex but simple with current technology. As the name may suggest, the background design requirement, it is not used to hide the radio signal but is employed because of other factors, not just processing gain but interference immunity, channel sharing and resistance to radio reflections (amongst others). It is therefore employed as security against operating conditions not for surveillance resistance. (Hedy Lamarr was a co-inventor and holds a patent for FHSS).

THE THINGS NETWORK

COMMUNITY

Communities
Start a community
Countries
Events
Marketplace
Responsible Disclosure

RESOURCES

Blog
Brand assets
Docs
Forum
Github
Labs
Learn
Map
Shop
Status

ABOUT US

Team
Contact
Professional
Press

SOCIAL

Facebook
Instagram
Linkedin
Angel
Twitter
Youtube

You are the network. Let's build this thing together.