La sécurité dans les réseaux LoRa

-

Projet de déploiement d'un réseau LoRa sur le campus de l'INSA

Henri Cazottes, Thomas Delmas, Audran Deyts, Gaël Loubet, Pierre Noël, Hélène Ravily
Tuteurs : M Alata et Mme Dragomirescu

Sources: LoRa Security Building a Secure LoRa Solution

Introduction

Dans le document étudié, il est indiqué "Le protocole LoRaWAN comme beaucoup de ses rivaux offre un chiffrement et des méthodes sécurisées pour mettre à disposition des nodes. Cependant ces fonctionnalités ne devraient pas être utilisées aveuglément par les développeurs et les utilisateurs puisqu'elles ne sont pas protégées contre toutes les attaques les prenant pour cible, d'autant que **leur efficacité est dépendante de l'implémentation faite par les développeurs**."

Nous tâcherons ainsi, en plus d'expliquer le fonctionnement de la sécurité LoRa, de lister les bonnes pratiques à suivre afin de garantir le réseau le plus sécurisé possible.

Contexte

Au cours de ce projet, nous nous intéresserons aux aspect de sécurité du réseau LoRa. Afin d'isoler les failles possibles auxquelles nous allons nous intéresser,

- nous considérerons que les gateway se situent dans des zones sûres et hors d'atteinte de tout public non autorisé et/ou malveillant (sur le toit d'un immeuble par exemple).
- nous nous intéressons à la classe A bidirectionnelle.
- -notre réseau a pour objectif de s'intégrer au réseau pré existant The Thing Networks. Le network server ne nous est pas accessible et nous le considérerons aussi comme "fiable".
- au contraire, les nodes seront localisés sur le campus et donc facilement accessibles par quiconque.

Au final, nous limiterons les modifications matérielles possibles comme suit:

- *Node* : Ajout / Modification / Retrait
- *Gateway*: Ajout

La sécurité dans LoRa

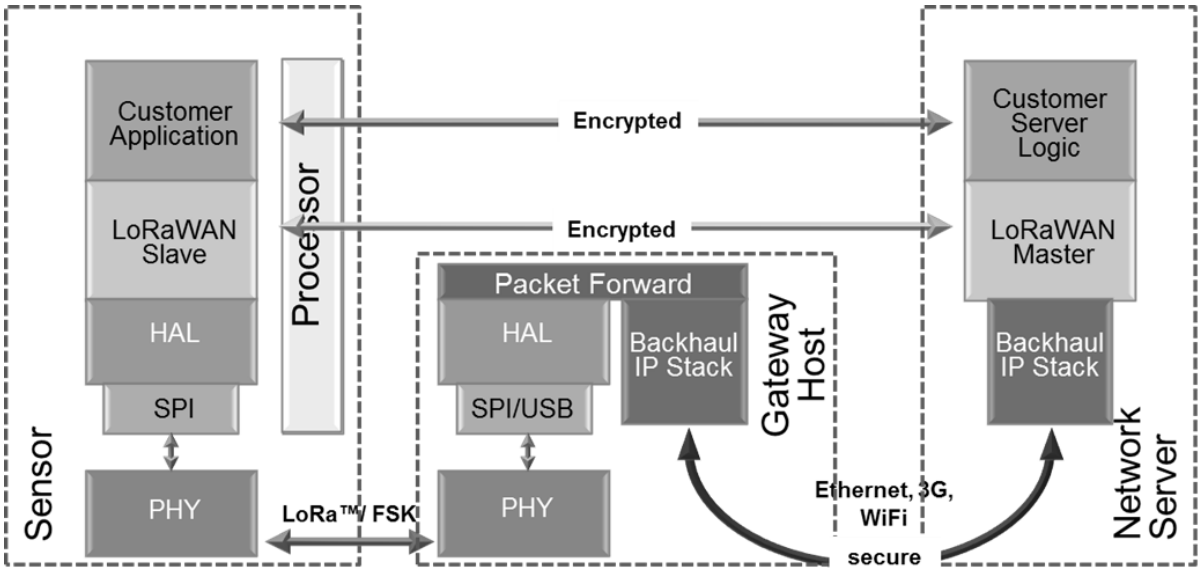
Activation: OTAA

Seuls le serveur et le node sont en connaissance de l'AppKey (clé de 128 bits).
Le message "join request" est signé en utilisant la l'AppKey et contient AppEUI le DEvEUI et un DevNonce. La signature est appelée MIC (Message Integrity Check) et est calculée de la manière suivante: MIC = aes(AppKey, MHDR | AppEUI | DevEUI | DevNonce)[0..3]. La "join request" est envoyée en clair.

Après avoir vérifié le MIC, le serveur envoie un "join accept" dans pendant la fenêtre temporelle d'écoute du node. Deux nouvelles clés sont calculées par le server:

- NwkSKey = aes128_encrypt(AppKey, 0x01 | AppNonce | NetID | DevNonce | pad16)
- AppSKey = aes128_encrypt(AppKey, 0x02 | AppNonce | NetID | DevNonce | pad16)

Les informations AppNonce, NetID et DevNonces sont envoyées au node (ainsi que plusieurs paramètres tels que le délai rf RxDelay et la liste de des channels utilisés CFList). Le join accept est chiffré à l'aide de l'AppKey.



Activation ABP

Le nœud contient déjà NwSKey et AppSKey et peut communiquer sans "join message".

Chiffrement

Les messages sont chiffrés selon l'algorithme AES128 en mode compteur. Si le FPort est à 0, le message ne contient que des informations liées à la couche MAC et la clé de chiffrement utilisée sera NwkSKey. Dans le cas contraire, la trame contenant des données applicatives sera chiffrée à l'aide de AppSKey.

Signature

La signature des messages permet de protéger la modification de paramètres tels que DevAddr, FCntUp ou FCntDown (les compteurs utilisés dans le chiffrement). Le payload MAC contient un MIC calculé de la manière suivante:

Msg = MHDR | FHDR | FPort | FRMPayload
*B0 = (0x49 | 4*0x00 | Dir | DevAddr | FCntUp or FCntDown | 0x00 | len(msg))*
mac = aes128_cmac(NwkSKey, B0 | msg)
MIC = mac[0..3]

Attaques envisagées et solutions

Node

Vol de la clé de chiffrement : la consommation du microcontrôleur effectuant le chiffrement peut être utilisée pour récupérer la clé. Le chiffrement étant symétrique il est ensuite possible de déchiffrer le contenu du message.

Déconnexion d'un nœud: lors d'un join request (après un reset hardware par exemple), le nœud va attendre une réponse de la part de la gateway lui indiquant certains paramètres de sessions. En brouillant de le médium pendant la fenêtre d'écoute, le nœud ne pourra pas se connecter à la gateway.

Usurpation d'identité: le nœud pouvant être accessible, s'il utilise un modem "haut niveau" qui implémente la stack LoRaWAN ainsi que la stack radio LoRa, il est possible d'envoyer des messages en se connectant directement sur le port série du modem.

Modification d'un message: on brouille la réception d'une gateway pendant qu'un nœud émet et on enregistre à la fois le paquet de ce nœud. On peut ensuite modifier le message de manière à l'altérer mais sans changer le MIC.

Gateway

DDOS: les gateway utilisées seront à base de raspberry pi, des ordinateurs ayant des capacités limitées. On peut envisager des dénis de service en floodant une gateway en particulier jusqu'à ce qu'elle ne puisse plus traiter la demande.

Man in the middle: Mise en place d'une gateway plus proche des noeuds cibles, ainsi les messages passeront par la gateway malicieuse puisque celle-ci aura un meilleur SNR.