Cédric DUVAL
IMA5

Rapport intermédiaire de Projet de Fin d'Etudes

P64 : Sécurité de l'IOT

Encadrants : Thomas VANTROYS
Alexandre BOE

Année 2016/2017

Sommaire

Présentation de l'IOT

L'internet des objets ou IOT (Internet Of Things), représente les échanges de données provenant d'appareils électroniques dans le monde réel vers le réseau Internet.

De nouvelles possibilités d'application de l'informatique à la vie quotidienne ou professionnelle s'ouvrent notamment dans le milieu de la domotique, de la santé ou du quantified-self (mesure de soi).

L'accroissement des objets connectés posent aussi le problème de la sécurité. En effet, on peut imaginer des détournements des fonctions premières des objets, et même des vols ou corruption de données pouvant entraîner des pertes pour les entreprises ou les particuliers utilisant ces objets.

Présentation du protocole LoRa

Le réseau LoRa (Long Range) est une technologie permettant aux objets connectés d'échanger des données de petites tailles. Le débit du réseau varie de 0,3 kb/s à 50 kb/s. La première utilité est que la consommation des objets est très faible garantissant une autonomie allant jusqu'à 10 ans. La portée maximale est d'environ 20km. L'architecture du réseau LoRaWan est utilisée sous forme de réseau hiérarchique où chaque passerelle peut transmettre les messages entre les appareils terminaux et un serveur de réseau central en arrière-plan.

Le protocole LoRa a 3 modes de fonctionnement :

- L'envoi d'informations vers une antenne puis la réception d'informations immédiatement après. Le serveur ne pourra envoyer d'informations qu'au prochain cycle d'envoi. Ce mode à la moins grande consommation d'énergie et permet d'envoyer des données de manière régulière (ex : capteur).

- La réception de données à intervalles réguliers et paramétrés à l'avance.

- La réception d'informations en continu, c'est le mode qui consomme le plus.

L'alliance LoRa a publié un programme de certification obligatoire consistant à vérifier que l'objet connecté répond aux contraintes fonctionnelles du standard LoRaWAN.

Les fréquences utilisées sont 433MHz, 868 MHz et 900 MHZ.

Cahier des charges

Comme nous n'avons pas accès au protocole LoRaWan, il faut créer une approche sécurisée qui permet d'envoyer un message et d'être sûr de sa bonne réception ainsi que de son intégrité. Il faut aussi s'intéresser aux problèmes de confidentialité. En effet, les informations peuvent être sensibles, et le fait de les altérer ou qu'un tiers puisse les lire peut mettre en danger l'entreprise ou le particulier l'utilisant.

Après avoir réalisé un protocole de sécurité, il faut créer une plateforme d'attaques permettant de tester la sécurité du protocole et la corriger si des failles sont présentes.

Enfin il faudra rédiger la documentation du protocole et de la plateforme afin que d'autres utilisateurs puissent s'en servir.

Récapitulatif :

Exigences pour le protocole :

- Authentification
- Confidentialité
- Disponibilité
- Intégrité

Plateforme d'attaques permettant de tester la sécurité

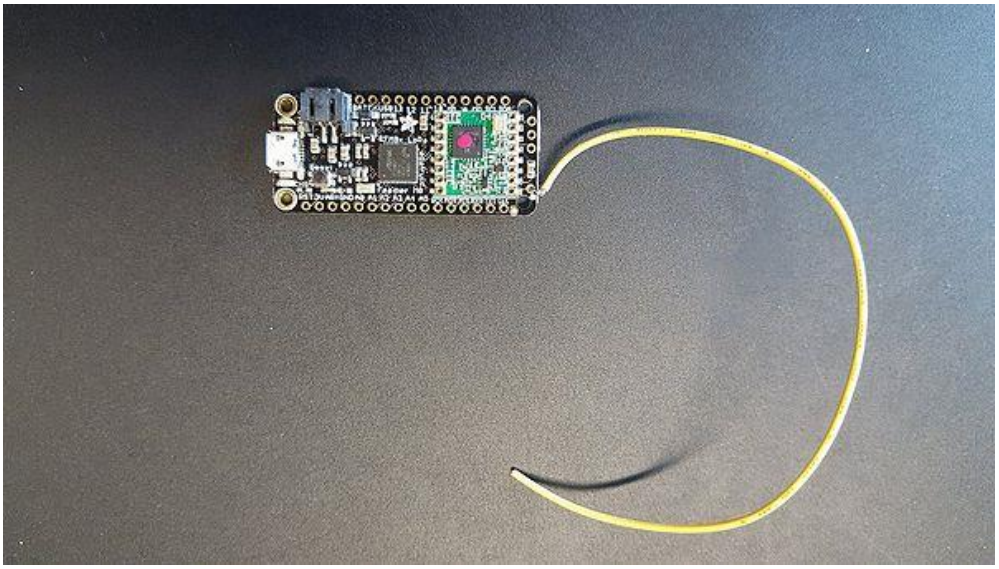
Rédaction de la documentation relative au protocole et à la plateforme

Présentation du travail effectué

Fonctionnement du module LoRa

Le module de communication LoRa utilisé pour ce projet est un **Adafruit Feather M0 RFM95 LoRa Radio**. Le module est équipé d'un processeur ARM CORTEX M0, contrairement à la plupart des Arduino. Sa portée d'émission, testée par le fabricant serait d'environ 2 Km avec une antenne fil. Le module peut être programmé avec l'IDE Arduino.

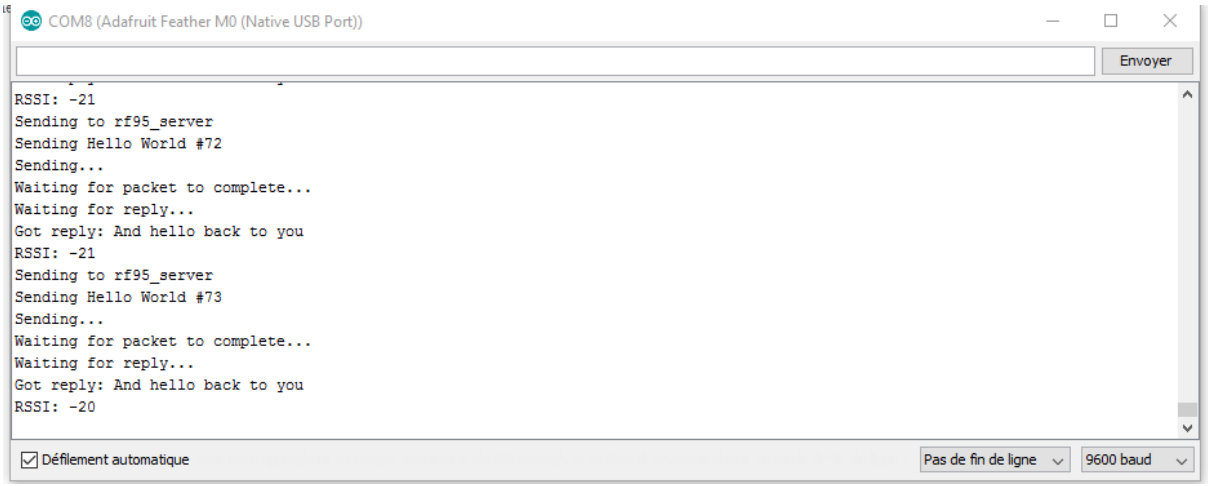
Afin de pouvoir tester le module, il a d'abord fallu souder un fil qui servira d'antenne. Le fil est à souder sur la pin « Ant. ».



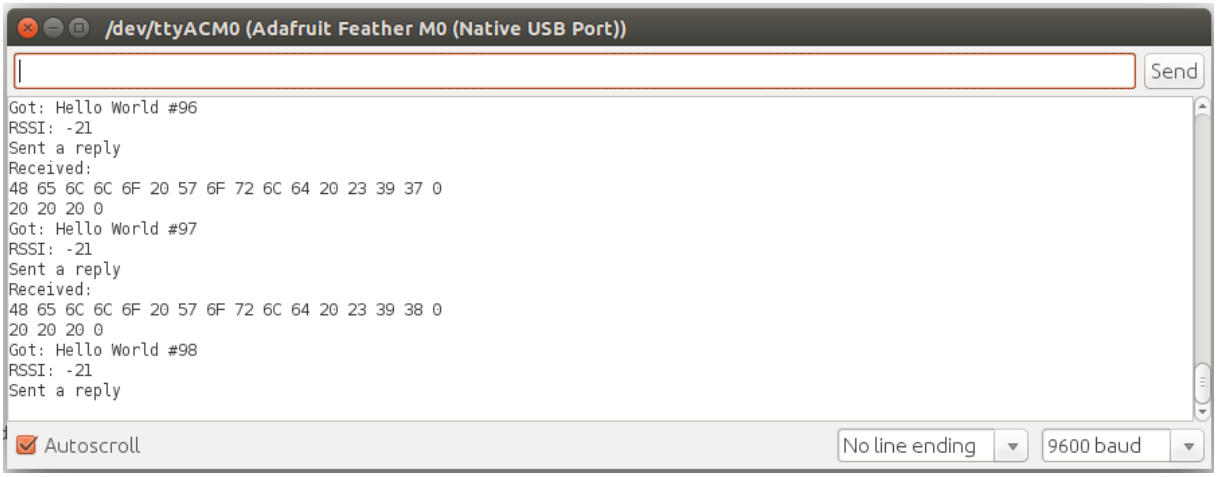
Une fois l'antenne installée, on peut tester la communication entre 2 modules.

Le code fournit par Adafruit permet d'envoyer un paquet d'un module à un autre, puis celui-ci répond au message envoyé.

Envoi du message :



Réception du message :



Les paquets envoyés avec programme ne sont pas sécurisés ni adressés et n'importe qui peut les récupérer en ayant un module LoRa réglé sur la bonne fréquence.

Depuis ce code, on peut en déduire plusieurs programmes permettant de mettre en place une approche sécurisée de la communication LoRa.

Fréquence de fonctionnement

Le premier point impactant la transmission entre les modules est la fréquence de fonctionnement. J'ai testé des valeurs différentes de celle prescrite par le constructeur soit 433 MHz. Les modules peuvent fonctionner de 390 à 550 MHz. Ces valeurs ne sont pas les limites de fonctionnement. Il faudra faire des tests plus précis de valeurs pour savoir les limites précises.

Un module doit normalement communiquer à la même fréquence que le récepteur. Cependant, on peut constater qu'en changeant la valeur de 0.02 MHz, la communication est toujours fonctionnelle.

Grâce à ces différentes données, on peut déjà établir une première mesure de sécurité consistant à attribuer aux modules différentes valeurs de fréquence, ou même changer leur fréquence de fonctionnement au cours du temps.

Les paquets peuvent être interceptés par n'importe quel autre module LoRa sur la même fréquence, un émetteur transmettant des paquets en continu peut se faire rapidement détecter.

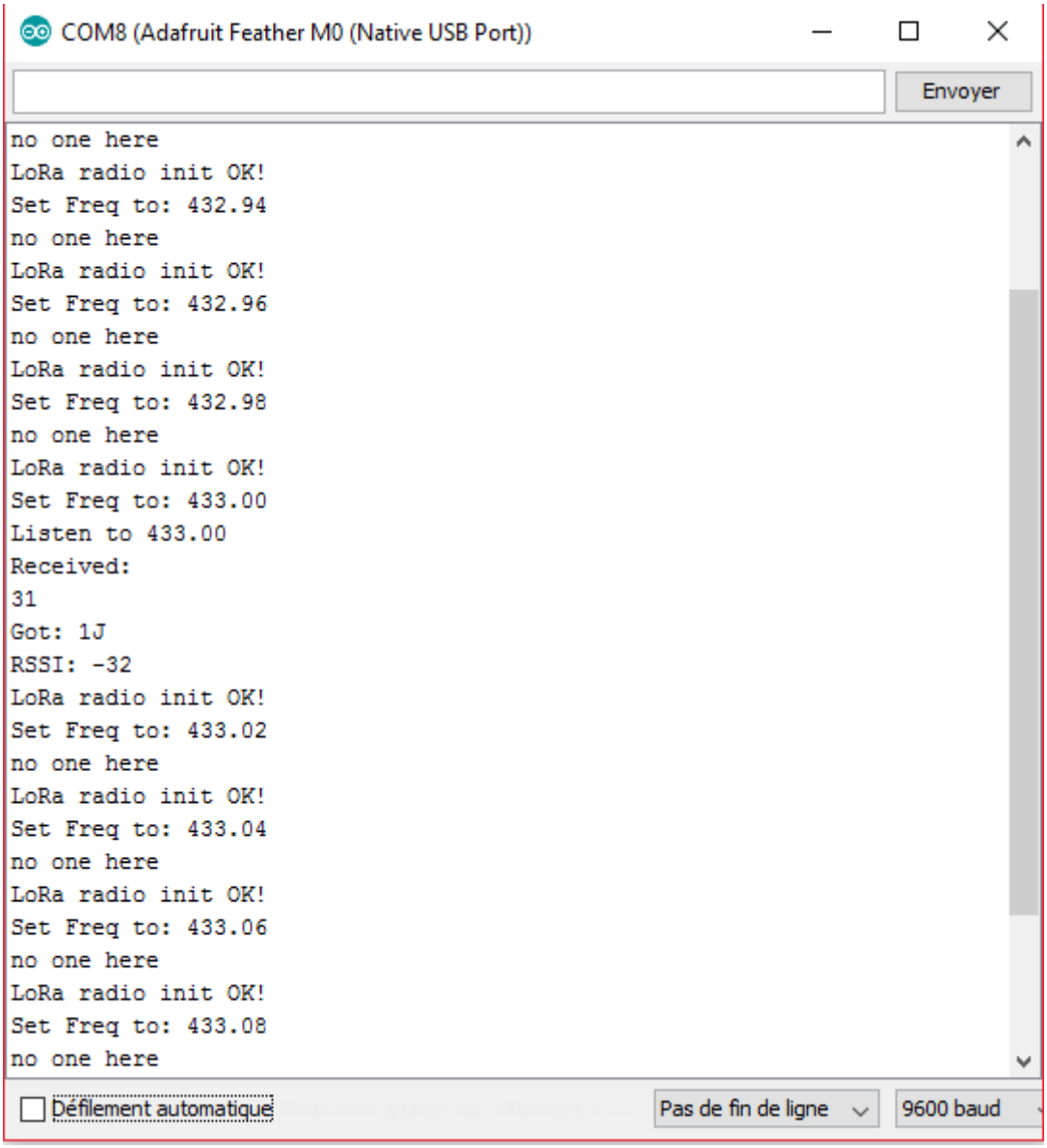
J'ai réalisé un programme qui scanne une bande de fréquence, permettant de savoir sur quelles fréquences des paquets sont émis. Le programme part d'une fréquence basse, et change la fréquence de son module LoRa de 0.02 MHz toutes les X secondes, suivant le temps disponible pour l'analyse. Les paquets reçus sont affichés et on peut voir pour quelle fréquence le paquet a été envoyé.

Ce genre de programme ne fonctionne pas bien lorsque les paquets ne sont pas transmis en continu, puisque l'on doit augmenter le temps d'attente pour chaque fréquence testée.

En attendant 3 secondes par fréquence, on scanne 1 MHz en 2m30s, ou 100 MHz en environ 4h. ce qui reste un temps assez long.

Des réglementations sont en place pour limiter les transmissions sur des canaux qui seraient réservés à des services publiques ou privés. Ainsi, la fréquence 433,05/434,79 MHz est limité à 10 mW de puissance en France et en Europe, soit10 dBm. La plage de fréquence 867 - 868,5 MHz contient 8 canaux de fréquences de 125 kHz mais je n'ai pas trouvé de réglementations ou de limites de puissance.

Scanner de fréquence avec paquet émis à 433 MHz :



Authentification

Afin que les paquets soient envoyés au bon module, il faut mettre en place un système d'adressage afin que les appareils puissent se reconnaitre entre eux. L'émetteur envoie le paquet avec l'adresse du destinataire. Les modules recevant le paquet regardent si le paquet contient leur adresse et l'ignore si ce n'est pas le cas. Si les adresses correspondent, le paquet est traité.

Ce système est simple mais tous les modules de réception recevront des paquets à chaque fois qu'un paquet est émis, même s'ils ne sont pas adressés à eux.

Disponibilité

Si un module est en panne ou rencontre un bug, il peut y avoir perte d'informations. Une solution pour éviter ce genre d'incident est de vérifier si le paquet est correctement reçu, si le récepteur est en panne. Une réponse du récepteur vers l'émetteur serait obligatoire pour s'assurer que l'émetteur n'envoie pas dans le vide.

Si l'émetteur n'a pas de paquet de confirmation, il envoie un paquet à un module dédié au report de panne pour lui informer que son paquet à destination de l'émetteur a rencontré un problème.

Si c'est l'émetteur qui rencontre un problème, le récepteur ne peut pas savoir qu'il est en panne à moins qu'il vérifie l'état de ses émetteurs toutes les X minutes. S'il n'y a pas de réponses à sa demande, il envoie un paquet au module dédié au report de panne.

Dans un cas plus simple où l'on pourrait faire fonctionner les modules assez fréquemment, on pourrait imaginer un système où tous les modules envoient un paquet au module de report de panne. Le module de report compare avec sa base de données de modules et indique lorsqu'un module ne lui a pas fait signe depuis plus de X minutes.

Le principal souci avec ces méthodes est si un brouilleur est mis en place, il serait difficile de le détecter, puisque le module chargé de recevoir les reports serait lui aussi brouillé.

Il faudrait donc que ce module puisse être relié de manière physique à un système de communication, car le module pourrait croire que les autres appareils sont HS mais que lui-même est toujours fonctionnel.

Intégrité des paquets

Les paquets doivent être vérifiés pour être sûr qu'ils ne soient pas corrompus. Lorsqu'un paquet est envoyé au récepteur, le récepteur renvoie le paquet à l'émetteur et celui-ci vérifie s'il est conforme à celui envoyé et renvoie le paquet s'il n'est pas conforme.

Pour améliorer le procédé et éviter plusieurs transmissions de paquets, un système de contrôle de somme peut être mis en place sur chaque paquet. Ainsi, le récepteur n'a plus à envoyer ce qu'il a reçu au transmetteur, réduisant le nombre de transmissions.

Un système de hachage peut aussi être incorporé sur le modèle SHA.

Confidentialité

Les paquets sont diffusés en clair et n'importe qui peut en voir les contenus, notamment les adresses. On pourrait facilement imaginer un programme qui intercepte les données, analyse les adresses et envoie aux adresses trouvées des paquets pour corrompre la base de données.

Il faut donc crypter les paquets afin que seul les modules disposant du moyen de les décrypter puissent être capables de comprendre les messages.

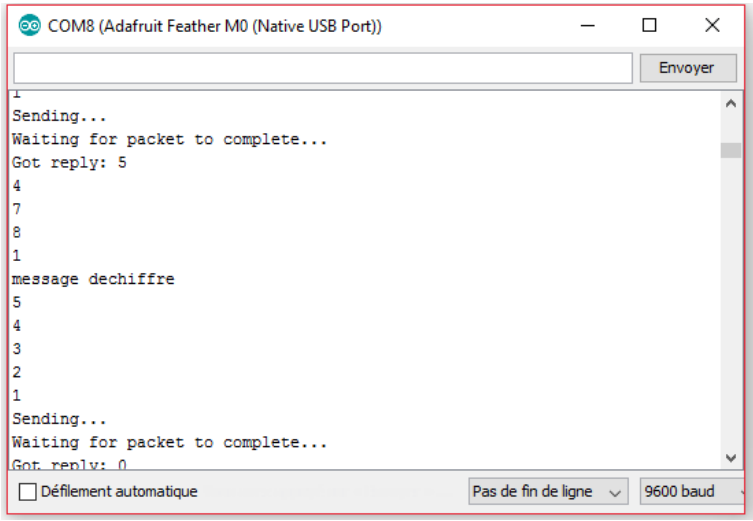
Je me suis orienté vers le cryptage RSA, car c'est le plus facile à mettre en place dans ce genre de communication, et qu'elle possède une sécurité très efficace.

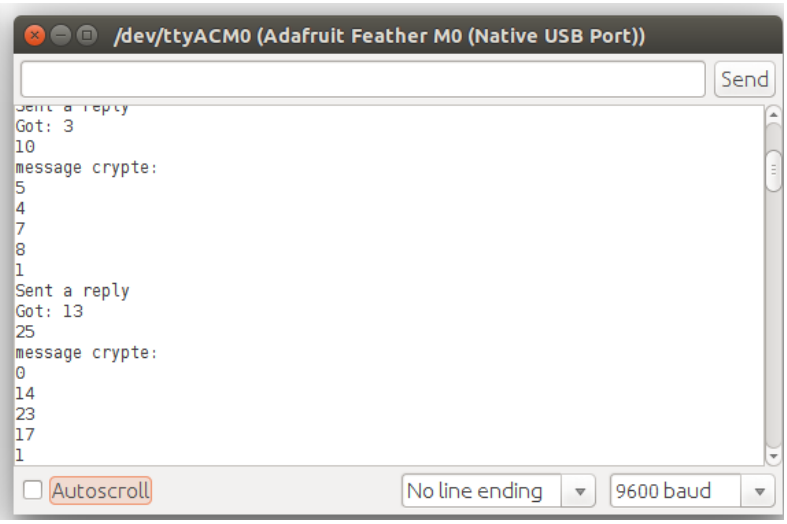
Le programme réalisé calcule aléatoirement une clé publique et privée. Le fait que la clé soit aléatoire et réinitialisée à chaque communication la rend plus sûre.

La clé publique est envoyée au module désirant communiquer. Celui-ci chiffre avec la clé les données et les envoie au récepteur. Ce dernier peut déchiffrer le message grâce à sa clé privée.

Le programme a encore quelques erreurs et quelque fois, le message déchiffré ne correspond pas aux données initiales. Ceci arrive environ 10% du temps, et varie en fonction des valeurs aléatoires données par le programme. La nécessité d'avoir un contrôle d'intégrité des paquets se révèle donc important.

Message des programmes émission et réception RSA :





Présentation du travail restant

Mise en place du protocole :

Le principe du protocole est de délivrer un paquet de manière sécurisée à un ou plusieurs modules. Pour cela il faut mettre en place une forme de paquet contenant l'adresse de destination, les données encryptées, puis la vérification d'intégrité du paquet.

En parallèle, un système de contrôle de disponibilité des modules doit être mis en place afin de garantir le bon fonctionnement de chacun des modules LoRa.

Plateforme d'attaques :

Une fois le protocole réalisé, il faudra tester sa sécurité en l'attaquant, le principe étant de l'améliorer au fur et à mesure que les attaques fonctionnent.

La plateforme consisterait en un site web hébergé sur une raspberry pi, elle-même connectée à un module LoRa. L'utilisateur pourrait lancer certains types d'attaque depuis le module et la plateforme renverrait les résultats.

Le fait que le site soit hébergé sur une raspberry pi rend la plateforme mobile, rendant plus facile les tests sur le terrain.

Documentation :

Il faudra documenter le protocole afin qu'il puisse être réutilisable. Il faut aussi créer un manuel d'utilisation de la plateforme d'attaques.