Sécurité LoRaWAN

Dossier de recherche

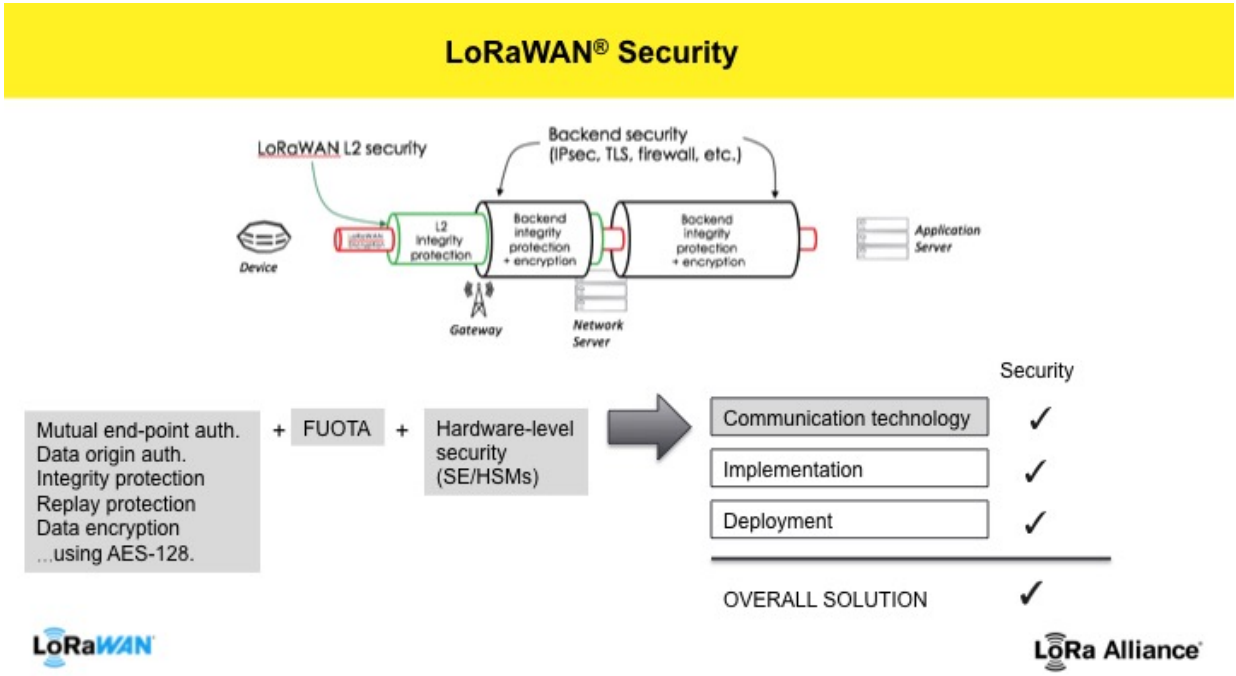
Chaigne Hyacinthe

2e semestre 2021

Contents

1 Introduction

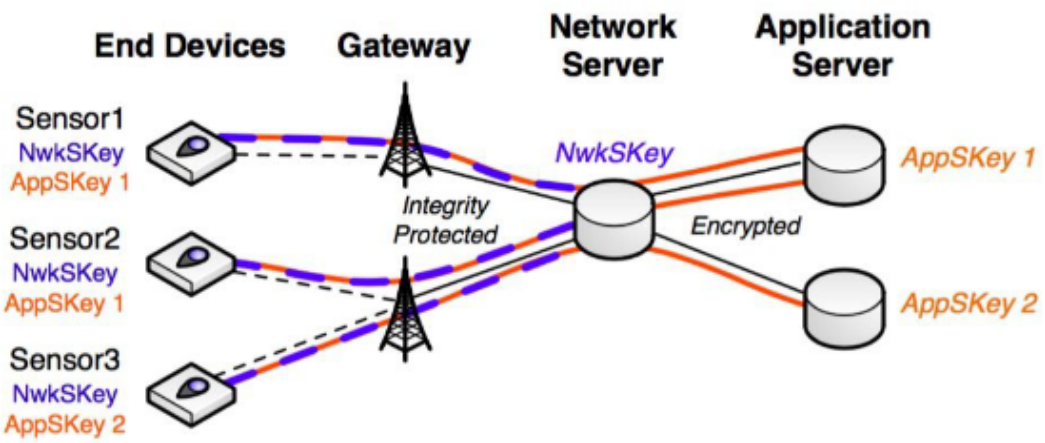
Basé sur des communications longue portée à bas débit et peu gourmandes en ressources, la technologie LoRa de *Semtech* à tout pour plaire dans le monde de l'IOT. Soutenue par Orange et Bouygues, elle tient tête à son concurrent français direct *Sigfox*. Se propageant à 868Mhz en Europe, celles-ci sont protégées par leurs clefs de chiffrement. En théorie, mais qu'en est-il lorsqu'une trame est capturée ?. Cet article à pour but de faire le point sur l'aspect sécurité du protocole et recenser les attaques possibles à ce jour. Le fonctionnement global des couches physiques, LoRaWAN et les systèmes de jointures sont jugés acquis pour comprendre cet article. Dans le contraire documentez vous suivant les explications sur lora-alliance et seeedstudio.



2 Versions protocolaire

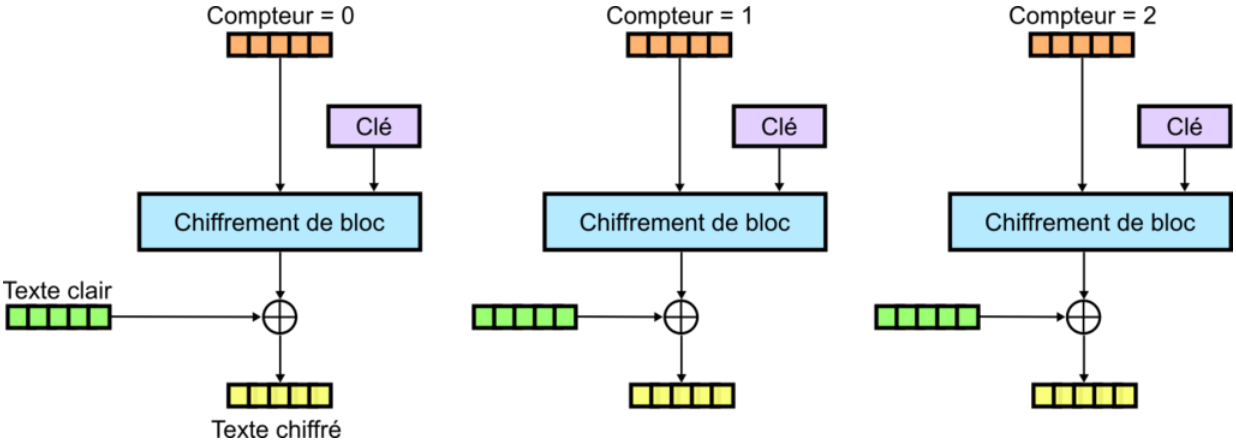
2.1 LoRaWAN 1.0

Que l'activation se fasse en mode ABP (Authentication By Personalisation) ou OTAA (Over The Air Authentication), la communication est sécurisée par deux clefs session AES-128 bits nommées **AppSKey** (Application key, entre l'appareil et un fournisseur d'applications tiers dans le backend) et **NwkSKey** (Network key, entre l'appareil et l'infrastructure du réseau). Ces clefs sont utilisées pour chiffrer les flux selon la méthode du **CCM** (Counter with CBC-MAC) étant lui même une variation de la méthode **CTR** (Counter Mode).



2.1.1 CTR

Le CTR est une méthode pour convertir un chiffrement par bloc (tel que AES128) en chiffrement de flux. Le texte à chiffrer est divisé en blocs de la taille définie par le chiffrement par bloc et chacun d'eux est combiné (ex: bit par bit par XOR) avec le numéro du bloc compteur dit **nonce** (nombre arbitraire destiné à être utilisé une seule fois) et d'une clé supposée secrète.



Ce mode permet de chiffrer de manière parallèle, c'est-à-dire chiffrer simultanément plusieurs blocs sur plusieurs processeurs pour accélérer le chiffrement d'un message. Le déchiffrement peut également être réalisé de manière parallèle.

2.1.2 CCM

Le CCM utilise le mode CTR pour chiffrer ses payloads en utilisant la clef **AppSKey** et authentifie les messages avec un code CMAC (Cipher-based MAC) basé sur la clef **NwSKey**. Les blocs de chiffrement (en turquoise) A_i sont chiffrés avec AppSKey ou NwkSKey en fonction du type de trame. Le résultat est utilisé avec les blocs de 128 bits et forment enfin le payload du message.

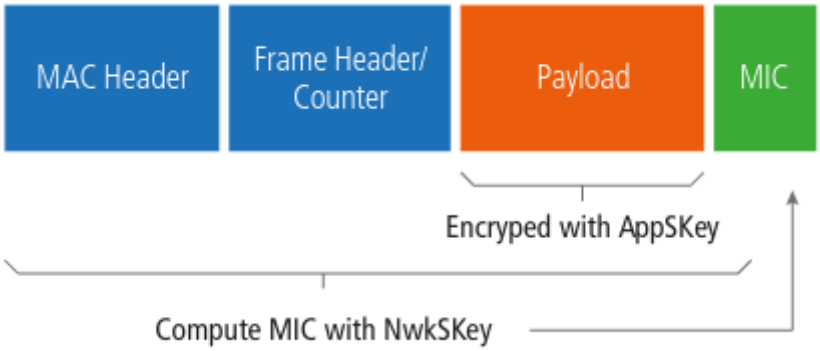
$$A_i = [1]_8[0]_{32}[D]_8[DevAddr]_{32}[C]_{32}[0]_8[i]_8$$
$$S_i = AES - 128(K, A_i)$$
$$TramePayloadChiffrée = [S_0][S_1]...[S_n] \oplus TramePayload$$

Les indices indiquent la taille de chaque champs concaténé en bits. Les octets des champs de plus de 8 bits sont assortis en **little endian** (ordre des octets dans lequel l'octet de poids le plus faible est enregistré à l'adresse mémoire la plus petite ou transmis premièrement)

- D : Adresse, 0 en liaison montante et 1 en liaison descendante
- C : Compteur de messages, 0 en début de session
- i : Le compteur de blocs répertorie le bloc AES auquel s'applique le résultat du chiffrement
- K : Clef NwkSKey ou AppSKey selon le type de trame MAC impliqué

Enfin, le résultat est authentifié selonc un CMAC AES de 128 bits afin de générer un MIC (Message Integrity Code) avec la NwSKey. Ce MIC est calculé sur le header, le payload chiffré et le Block B_0.

$$B_0 = [0x49]_8[0]_{32}[D]_8[DevAddr]_{32}[C]_{32}[0]_8[TailleHeader + TaillePayload]_8$$
$$MIC = AES - 128 - CMAC(NwkSKey, [B_0][Header][TramePayloadChiffrée])$$



Les propriétés matérielles sont :

- Sens de la trame (montant/descendant)
- Appareil avec adresse 32 bits
- compteur de messages
- Une clef AES de 128 bits

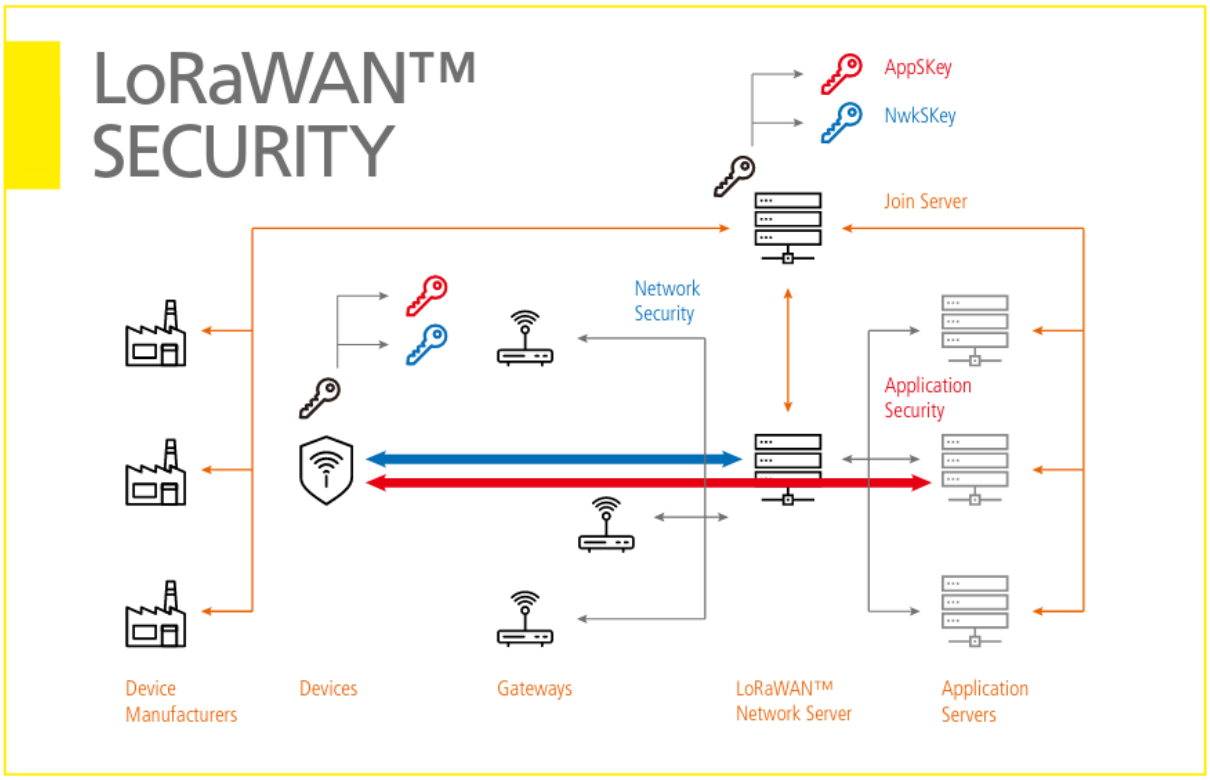
2.1.3 Bilan

Comme vu plus haut le protocole LoRaWAN 1.0 **réutilise les compteurs de messages** pour chiffrer ses trames en réinitialisant son compteur à la même valeur. Il faudrait en théorie un très grand nombre de trames afin de déterminer ses cycles et casser le chiffrement. Notons qu'un layer supplémentaire de chiffrement de bout en bout aurait pu garantir l'intégrité de la trame transmise. L'adresse AppKey étant partagée par la gateway et les serveurs applications, rien ne garanti une interception de trames entre deux. Dans larticle "Security of LoRaWAN v1. 1 in Backward Compatibility Scenarios" de Dönmez, T.C.; Nigussie, E. Les vulnérabilités suivantes ont été trouvées :

Durant le processus de join des cartes, **la génération est utilisation des nonces n'est pas bien intégrée**, on pourrait imaginer réutiliser ces nonces. **La gateway n'est pas capable de détecter l'envoi répété de nonces durant un join OTAA**. **Le serveur réseau ne détecte qu'un certain nombre N non spécifique de DevNonces**, un attaquant ne doit attendre que N messages avant de rejouer un message de demande de join. **Les messages envoyés ACK ne sont pas associés à de messages spécifiques**, il est donc possible de spammer. De même pour les messages de join.

2.2 LoRaWAN 1.1

Même si ces failles semblent inexploitables pour certaines, elles sont bien réelles. Afin d'éviter de potentiels exploits naître le protocole se mets à jour avec des mesures renforcées. Ces mesures rendent donc les protocoles 1.0 et 1.1 incompatibles à la communication si l'appareil ne supporte pas le 1.0 (le marché suit la nouvelle version). Un serveur peut faire tourner les deux versions, en théorie aujourd'hui un serveur n'est sécurisé que s'il n'exécute exclusivement du 1.1.
Retrouvez les changements protocolaires à cette adresse, la vidéo est présentée par un membre de The Things Network et couvre l'intégralité des modifications. Considérez également retrouver la liste des spécifications des versions publiées. Selon cet article les versions 1.0.x sont tenues à jours (1.0.4, 28 octobre 2020) afin de garder les appareils incompatibles à la v1.1.x les plus sécurisés possibles.



3 Types d'attaque

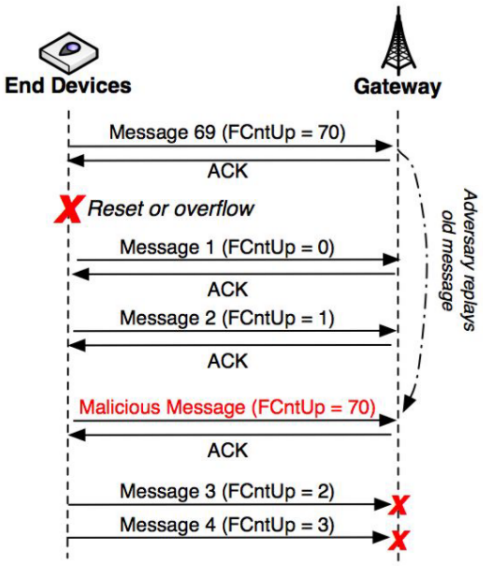
Parce que LoRaWAN fonctionne avec un système de canaux, ceux-ci peuvent par défaut être sujet à des injections. Faut-il que la trame injectée soit recevable et provenir d'une source légitime. Devenant nécessaire d'échanger une paire de clefs à un instant T pour appareiller. Voici une liste des attaques possibles sur le protocole 1.0.x.

3.1 Replay attack

Cette attaque vise les noeuds paramétrées en mode ABP. Pour rappel les deux clefs sont statiques et programmées sur l'appareil. En conséquence, ces appareils **réutilisent le compteur de trames** à 0 avec ces deux mêmes clefs quand le compteur **overflow**.

En tant qu'attaquant il nous faudrait analyser et stocker tous les messages envoyés. Attendre que le compteur se reset et rejouer un message au bon moment. Faire ceci en boucle bloquerait l'appareil comme une attaque DoS (Denial of Service).

Générer périodiquement deux nouvelles clefs ou adopter le mode OTAA pallierai cette vulnérabilité.



3.2 Eavesdropping

Comme sa définition l'indique, l'objectif est d'écouter une conversation sensée secrète. En reprenant le principe de la précédente attaque et son compteur nous sommes capables de **prédire les blocs de chiffrement à chaque fois que celui-ci overflow**.

Donné la relation $P \oplus K = C$

- P : Texte
- K : Bloc de chiffrement
- C : Texte chiffré

Sur deux textes chiffrés $P_1 P_2$ sous le même bloc de chiffrement K nous aurions :

$$C_1 \oplus C_2 = (P_1 \oplus K) \oplus (P_2 \oplus K)$$
$$C_1 \oplus C_2 = (P_1 \oplus P_2) \oplus \underbrace{(K \oplus K)}_{0}$$
$$C_1 \oplus C_2 = (P_1 \oplus P_2)$$

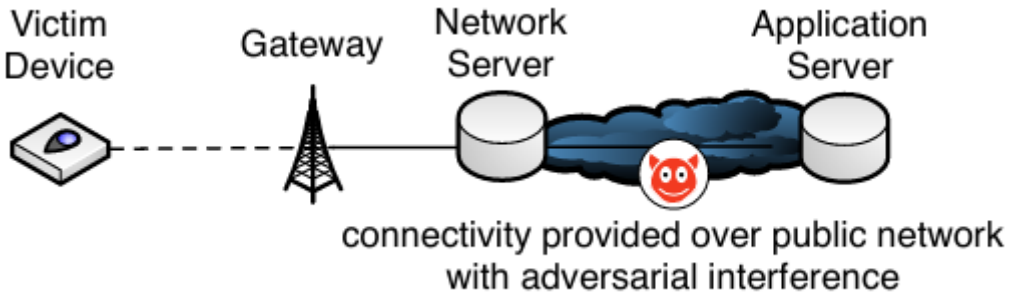
Table de vérité de XOR		
A	**B**	**R = A ⊕ B**
0	0	0
0	1	1
1	0	1
1	1	0

Par défaut cette méthode fonctionne si **l'overflow reset le compteur à 0**, l'utilisation d'une valeur déterminée par un **nonce rendrait l'attaque bien plus complexe**.

3.3 Bit-flipping attack

Alors que les messages sont à la fois chiffrés et équipés d'un contrôle d'intégrité, les deux fonctionnalités ne sont pas appliquées à la même portée. D'après la description du protocole, le code d'intégrité du message chiffré sur le payload et les informations du header sont vérifiés et terminés par le fournisseur d'infrastructure réseau, tandis que le chiffrement du payload à l'aide de l'AppSKey est annulé par le fournisseur d'application. Cela signifie qu'entre le serveur réseau de l'opérateur de l'infrastructure et le serveur d'applications du fournisseur, **l'intégrité et l'authenticité du contenu ne peuvent pas être vérifiées**.

Alors qu'un chiffrement par bloc réagit normalement aux modifications binaires, le principe de l'effet d'avalanche qui rendrait normalement illisible un message inversé de bits... AES n'est uniquement utilisé comme générateur de flux de clé et ce chiffrement est facilement malléable à moins qu'un contrôle d'intégrité ne soit combiné avec le déchiffrement . Cependant, LoRaWAN 1.0.2 s'écarte des pratiques recommandées. Laissant une porte à un injection.



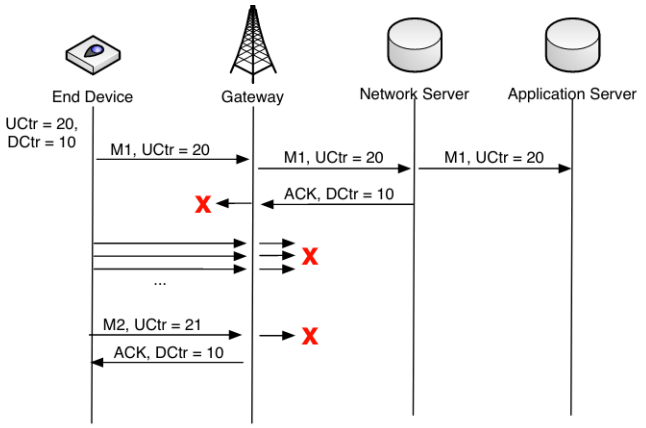
3.4 ACK spoofing

Pour maximiser la durée de vie de la batterie, le mécanisme d'accusé de réception des données est rendu facultatif. Admettons la réponse ACK suivante, le message ne spécifie pas à quel message il accuse réception.

PHYSICAL PAYLOAD FORMAT OF AN ACK MESSAGE.

MHDR	DevAddr	FCtrl	FCnt	MIC
60	88889999	20	0B00	BAE1557A

Le MIC confirme bien l'authenticité du message, le compteur (FCnt) est lui séquentiel par rapport à tous les messages descendants. Capturer l'une de ces réponses ACK pourrait permettre a un attaquant de retarder les validations et sélectivement valider un autre message sans rapport, même s'il n'est pas arrivé à destination. Afin de réaliser cette attaque, il faudrait jam les bandes passantes près de la gateway et capter toutes les trames "envoyées" depuis le cache de l'appareil.



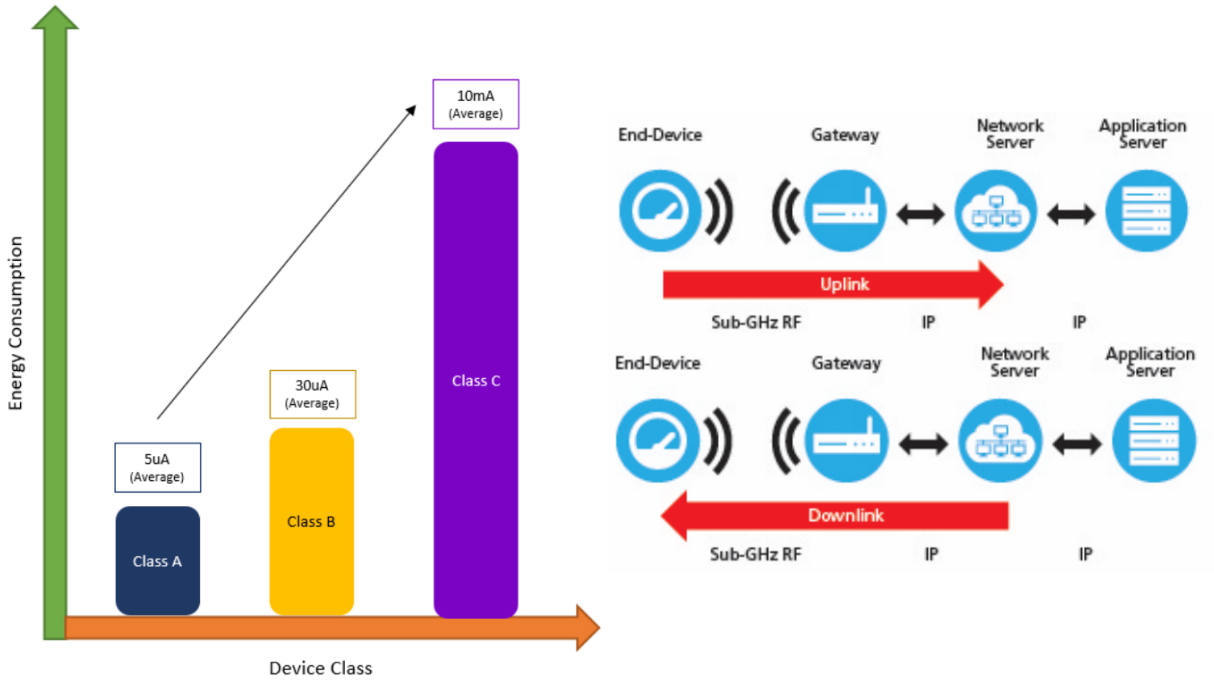
3.5 LoRa class B attacks

3.5.1 Rappel

Les nœuds de **classe A** passent la plupart de leur temps en mode veille. LoRaWAN n'est pas un protocole à fente, les nœuds peuvent communiquer avec le serveur réseau à chaque fois qu'il y a un changement dans la lecture d'un capteur ou lorsqu'un temporisateur se déclenche. Fondamentalement, ils peuvent se réveiller et parler au serveur à tout moment. Une fois que l'appareil envoie une liaison montante, il écoute un message du réseau pendant une et deux secondes après la liaison (des fenêtres de réception) avant de se rendormir. Cette classe est la plus économe en énergie, l'inconvénient est qu'un échange n'est possible que lorsque qu'un noeud décide de communiquer.

Plutôt que d'attendre seulement que l'un de ses capteurs remarque un changement dans l'environnement ou déclenche une minuterie, les nœuds de **classe B** se réveillent également et ouvrent une fenêtre de réception pour écouter une liaison descendante selon un calendrier défini par le réseau. Un signal périodique transmis par le réseau permet à ces nœuds de synchroniser leurs horloges.

En **classe C**, les noeuds écoutent constamment les messages de liaison descendante du réseau, sauf lors de la transmission de données en réponse à un événement de capteur. Ces appareils sont plus gourmands en énergie et nécessitent généralement une source d'alimentation constante, plutôt que de dépendre d'une batterie.



3.5.2 Attaque

Contrairement aux autres attaques, celle-ci est spécifique à la classe B. Elle permettrait à un attaquant de drainer toute la batterie des dispositifs en classe B. Comme expliqué, cette classe a pour objectif d'être un juste milieu entre la consommation d'énergie et la possibilité des périodiquement réceptionner des liaisons descendantes. le rythme est défini par l'envoie de trames régulières pour aligner les horloges des noeuds.

BCNPayload	NetID	Time	CRC	GwSpecific	CRC
Size (bytes)	3	4	1	7	2

Ces trames ne sont ni chiffrées, ni protégées à la modification. Puisque que c'est trame est diffusée en broadcast, un attaquant pourrait récupérer et modifier ces trames pour forcer les noeuds à être actifs en permanence. Vidant ainsi leurs batteries comme s'ils fonctionnaient en classe C.

4 Crack it

4.1 Récupérer le payload

La première étape vise à écouter le trafic LoRaWAN qui circule, il faut donc créer un "packet sniffer" qui automatisera la tâche. Ce turoriel ou celui-là proposent une manière simple de setup ce genre d'appareil (sans déchiffrement). Notez que les bandes de fréquence en Europe sont comprises entre 863 Mhz et 870 Mhz.

4.2 Déchiffrer

Dans le cas où nous venions à réceptionner un payload et obtenir les clefs NwkSKey et AppSKey d'un noeud. Il existe bon nombre de décodeurs tels que lorawan-packet-decoder utilisable en ligne ou de petits scripts pythons comme python-lora. La page couvre les modes ABP et OTAA et explique les étapes à suivre.

5 Conclusion

Sécuriser le déploiement de l'IoT et le garder ainsi ne consiste pas seulement à choisir le bon protocole, il repose sur le processus de sa mise en œuvre, de bonnes pratiques et de normes industrielles.

LoRaWAN est de par sa conception très sécurisé à savoir l'authentification et le chiffrement. Cependant tout réseau et appareil peut voir son système compromis si les clés venaient à être partagées, où si elle n'étaient pas générées aléatoirement ou si les numéros cryptographiques utilisés (nonces) étaient réutilisés. C'est pourquoi il est essentiel de rechercher des périphériques LoRaWAN CertifiedCM pour assurer le respect des normes du périphérique.

Par accès physique aux appareils, des méthodes de reverse engineering sur le firmware permettrai d'extraire ces clefs. En janvier 2020, beaucoup d'appareils déployés se voyaient fournis avec un QR-code récapitulant leurs informations. Certains généraient leurs clefs selon *AppKey = device identifier + app identifier* ou *AppKey = app identifier + device identifier*. D'autant plus qu'un appareil mal configuré ouvre une porte supplémentaire aux hackers.

La sécurité de l'IOT est en constante évolution et représente l'un des piliers de la *Smart City* de demain. La montée en popularité de LoRa verra ses acteurs malveillants augmenter et traiter des potentielles nouvelles menaces en amont servira de contrat de confiance auprès de la communauté. Proposer des solutions faciles d'utilisation et propices est un atout non négligeable à leurs concurrent *Sigfox* qui envoie ses payloads en clair.

6 Sources

https://www.includehelp.com/cryptography/counter-ctr-mode-in-cryptography.aspx

https://openclassrooms.com/fr/courses/1757741-securisez-vos-donnees-avec-la-cryptographie/6031865-utilisez-le-chiffrement-symetrique-pour-proteger-vos-informations

https://www.tarlogic.com/en/blog/cybersecurity-in-lora-and-lorawan-context-and-background/

https://lora-alliance.org/search/specification

https://www.tarlogic.com/en/blog/lorawan-1-0-vulnerabilities-and-backward-compatibility-in-version-1-1/

https://www.01net.com/actualites/objets-connectes-les-reseaux-lorawan-vulnerables-aux-attaques-de-hackers-1042538.html

https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_security_whitepaper.pdf

https://www.cyber-threat-intelligence.com/publications/IoTDI2018-LoraWAN.pdf

https://lora-alliance.org/resource_hub/lorawan-is-secure-but-implementation-matters/

https://www.zdnet.com/article/lorawan-networks-are-spreading-but-security-researchers-say-beware/

https://lora-developers.semtech.com/library/tech-papers-and-guides/lorawan-class-a-devices