Université de Mons
Faculté des Sciences
Département d'Informatique
SERVICE RESEAU ET TELECOMMUNICATION

Identifying IoT nodes by spying on their radio signal

Directeur : M^me^ Bruno QUOITIN

Rapporteurs : M^r^ Prénom NOM
M^r^ Prénom NOM

Mémoire réalisé par
Arnaud TULIPPE HECQ

en vue de l'obtention du grade de
Master en Sciences Informatiques

Année académique 2023-2024

Remerciements

Nous remercions ...

Table des matières

Introduction

L'avènement de *L'Internet of Things* (IoT) a lancé une nouvelle ère d'appareils connectés, ouvrant de nouvelles possibilités de partage de l'information, d'automatisation et de protection. Bien que le concept lui même soit prometteur, la technologie qui l'accompagne est essentielle. Les premières technologies utilisées pour l'IoT étaient les technologies sans fil déja présentes comme le Wifi ou le Bluetooth. Elles ont cependant plusieurs limitations : une consommation en énergie élevée, une portée restreinte et parfois même un coût d'infractrucutre trop important. Dans ces circonstance est apparu *LoRa*, une technologie développée en particulier pour l'IoT. Sa capacité à gérer les communications longue portée même dans des environements urbain très dense est un grand atout pour le domaine.

L'expansion de L'IoT soulève une nouvelle problèmatique de sécurité. Entre autres, l'identification des noeuds au sein des réseaux est essentielle. Il a été découvert que des noeuds fabriqués avec les mêmes microprocesseurs et modèles d'émetteurs-récepteurs radio peuvent présenter de subtiles particularités dans les caractéristiques de leurs signaux. Cette variabilité intrinsèque de la transmission des signaux radio peut être exploitées pour distinguer les noeuds d'un réseau. En écoutant leurs signaux radio et en analysant leurs signatures distinctes, il devient possible de les identifier.

Ce travail est structuré en quatre parties. Le premier chapitre sert d'aperçu global du signal radio afin d'y développer et rappeler les concept de télécommunication de base. Ce chapitre présente également les technologies LoRa et LoRaWAN à travers leurs caractéristiques et leur pertinence dans l'IoT. Le second chapitre rassemble les travaux qui ont déja été effectués dans ce domaine. Le troisième chapitre est dédié à l'étude expérimental du sujet. Les aspects pratiques y seront appliqués, notamment l'utilisation de radio logicielle afin de capturer des signaux radio. Ces signaux seront ensuite analysés grâce à diverses méthodes détaillés dans ce chapitre. La dernière partie du travail présentera les résultats obtenu en suivant l'analyse effectuée au chapitre précédant. Enfin le travail sera achevé en concluant sur de potentielles implications plus larges à ce sujet ainsi que des recherches plus approfondies.

Chapitre 1

Rappels et nomination des technologies

1.1 Signal radio

Un signal est une variation dans l'espace ou dans le temps d'une quantité physique contenant de l'information. Un signal peut être continu ou discret, on le nomme alors respectivement analogique ou numérique. Le type de signal dépend notamment de l'information qu'il contient. Un signal analogique est continue en amplitude, ce qui veut dire qu'il peut contenir un nombre infini de valeur, ainsi que prendre toutes les valeurs possibles, là où un signal numérique contient généralement un nombre fini de valeur (par exemple des 0 et 1). Les deux catégories ne sont pas incompatible car il est souvent nécessaire en télécommication de pouvoir passer de l'un à l'autre.

L'utilisation de signaux radio en télécomunication confère de nombreux avantages, comme la portée, la vitesse de transmission ou encore le coût de propagation. Pouvoir transporter de l'information sans avoir recours à du support matériel complet (pas besoin de cable, le signal passe dans l'air) réduit considérablement le cout de la transmission. Ajouté à cela, il est possible d'adapter un signal pour le rendre compatible avec diverse canaux de transmission et de réception, grâce à la modulation. La modulation est une technique permettant de modifier les propriétés du signal lui permettant de transporter de l'information.

En télécommunication, les signaux sont des ondes électromagnétiques appelées signal radio. Les signaux comportent de nombreuses caractéristiques qui les déterminent :

la fréquence, mesurée en Hertz (Hz). Elle détermine le nombre de cycle qu'accomplit le signal par seconde. Une onde radio possède une fréquence entre 9kHz et 300GHz.

La largeur de spectre, elle dépent de la fréquence car c'est l'écart entre la plus haute et la plus basse fréquence du signal. Une plus grande largeur permet de transmettre plus d'informations, mais consomme plus d'énergie.

L'amplitude. Selon le type de signal l'attribut possède différentes fonctions. Dans le cas d'un signal analogique l'amplitude détermine la magnitude de l'onde pour n'importe quel point dans le temps. Dans un signal numérique l'amplitude est interprétée différemment. Les signaux numériques sont encodés avec des valeurs discrètes, où chaque valeurs repésente un niveau (par exemple 0 ou 1). L'amplitude permet de faire la disctinction entre ses niveaux.

la puissance, mesurée en Watt (W). C'est la force du signal, un attribut important pour la réception du signal notamment. Bien que le Watt soit utilisé pour décrire la puissance à l'émission ou la réception, les variations de puissances sont généralement exprimées en décibels (dB). Le décibel est une unité logarithmique permettant de mesurer plus facilement les realtions entre les différents niveaux de puissance.

le *Signal to Noise Ratio* (SNR). Cet attribut mesure la qualité du signal. une valeur élevée indique que le pourcentage de bruit est faible.

le *Bit rate*, ou le taux de transmission mesure la quantité de donnée transmise en bit par seconde. Cet attribut est exclusif aux signaux numériques. On parle de *Baud rate* pour mesurer la quantité de symboles transmise par seconde. Ce n'est pas excatement l'équivalent du bit rate car un symbole peut contenit plusieurs bits, mais le Baud rate est utilisé pour les signaux numériques et analogiques.

1.2 Traitement du signal

1.2.1 Modulation

La réception d'un signal radio nécessite une antenne dont les dimensions dépendent de la longueur d'onde du signal. La longueur d'onde d'un signal représente la distance entre deux points consécutifs de même phase dans l'onde. La longueur d'onde s'obtient par la formule suivante :

$$c = f * \lambda \tag{1.1}$$

où c est la vitesse de la lumière,
f est la fréquence,
λ est la longueur d'onde.

Il est donc possible d'adapter les charactéristiques d'un signal pour le rendre compatible à différentes antennes, via la modulation. La modulation est le procédé par lequel un ou plusieurs attributs du *baseband signal* (le signal modulant, contenant l'information à transmettre) vont être altéré par le *carrier signal* (un signal porteur, utilisé pour être combiné avec le signal modulant) pour devenir un signal modulé, le *modulated signal*.

En plus de sa compatiblité, un signal modulé a l'avantage d'être facilement transmissible sur une grand portée sans perdre en puissance.

Parmis ces différents attributs, certains sont utilisés pour effectuer une modulation. Les trois modulations les plus utilisés sont basées sur les attributs de la fréquence, l'amplitude et la phase. La modulation en fréquence (*Frequency modulation, FM*) consiste à encoder l'information en faisant varier la fréquence en maitenant l'amplitude constante. La modulation en amplitude (*Amplitude modulation,AM*) est le procédé inverse, c'est à dire encoder l'information en faisant varier l'amplitude tout en gardant la fréquence constante. La modulation en phase (*Phase Modulation, PM)* fait varier la phase de la porteuse proportionellement à l'amplitude instantannée du baseband signal.

La modulation en amplitude est plus ancienne et est encore utilisée dans beaucoup de systèmes. Cette technique possède moins de contrainte et est notamment plus simple à implémenter.

Soient $u(t)$ un baseband signal et $v(t)$ un carrier signal, la modulation en amplitude s'effectue en multipliant les deux signaux pour obtenir le signal modulé

$$s(t) = u(t).v(t) \tag{1.2}$$

Prenons par exemple
$u(t) = sin(2\pi f_u t)$ avec $f_u t$ = 5 Hz,
$v(t) = cos(2\pi f_c t)$ où $f_c t$ = 50 Hz.
La Figure 1.1 montre le signal modulé $s(t)$ via la modulation en amplitude.

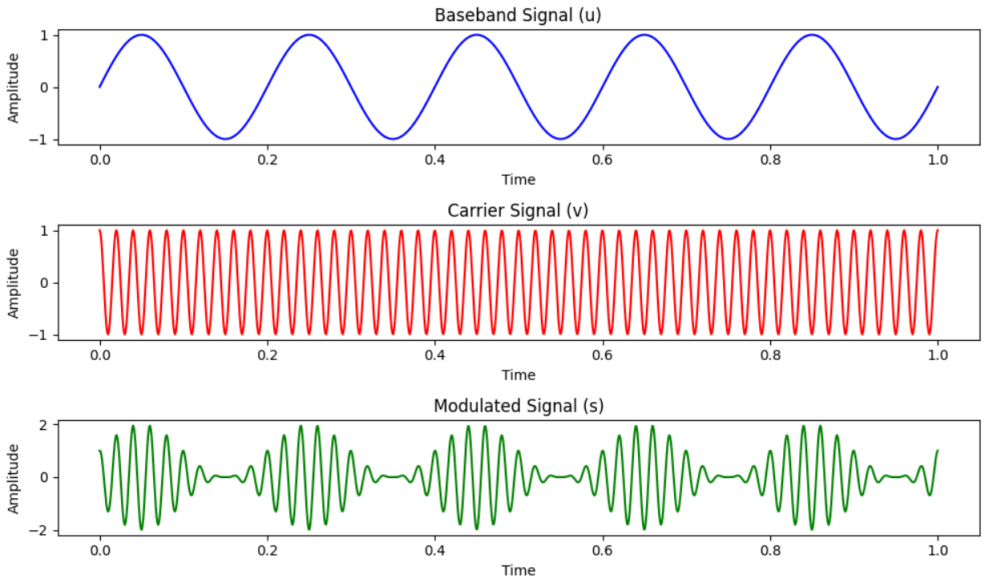

FIGURE 1.1 – Exemple de modulation en amplitude

La modulation en fréquence permet d'obtenir des transmissions de meilleures qualités plus résistantes à leur environement tout en gardent une puissance d'émission constante.

Soient $u(t)$ un baseband signal et $v(t)$ un carrier signal, le signal modulé en fréquence $s_{fm}(t)$ est le résultat suivant :

$$u(t) = \sin(2\pi f_u t) \tag{1.3}$$
$$v(t) = \cos(2\pi f_c t + \phi_c) \tag{1.4}$$
$$s_{fm}(t) = \cos\left(2\pi f_c t + \Delta f \cdot u(t) \cdot t + \phi_c\right) \tag{1.5}$$

Prenons par exemple

$u(t) = sin(2\pi f_u t)$ avec $f_u t$ = 5 Hz,
$v(t) = cos(2\pi f_c t + \phi_c)$ où $f_c t$ = 50 Hz.

La Figure 1.2 montre le signal modulé $s_{fm}(t)$ via la modulation en fréquence pour une phase initale ddu carrier signal ϕ_c = 0 avec une dérivation en fréquence Δf = 10Hz.

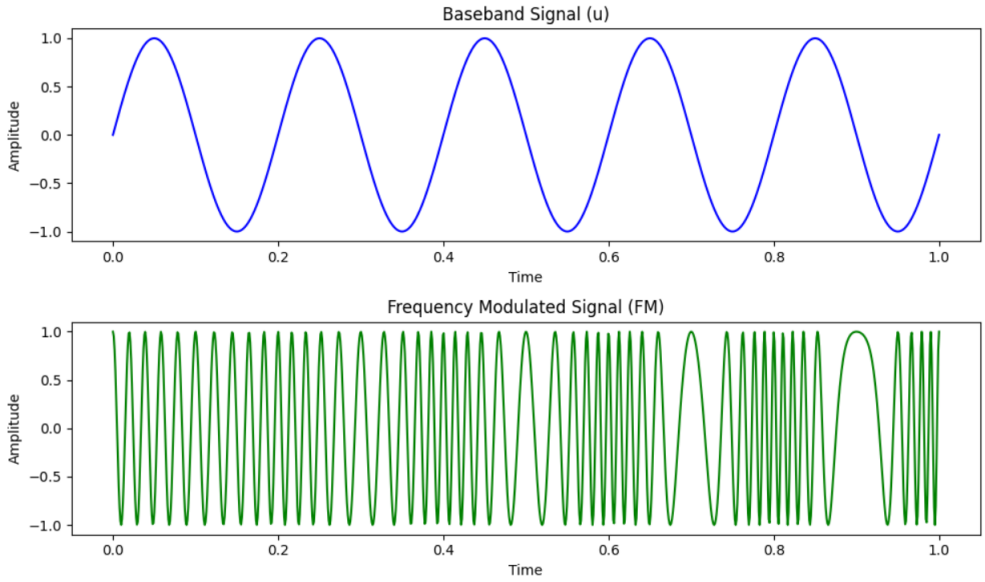

FIGURE 1.2 – Exemple de modulation en fréquence

La modulation en phase permet généralement d'obtenir une meilleur utilisation de la bande passante que les autres modulations car les variations de phase peut encoder plus d'informations, ce qui augmente la quantité de données transmises.

Soient $u(t)$ un baseband signal et $v(t)$ un carrier signal, le signal modulé en phase $s_{pm}(t)$ est le résultat suivant :

$$u(t) = \sin(2\pi f_u t) \tag{1.6}$$
$$v(t) = \cos(2\pi f_c t + \phi_c) \tag{1.7}$$
$$s_{pm}(t) = \cos\left(2\pi f_c t + K_p \cdot u(t)\right) \tag{1.8}$$

Prenons par exemple

$u(t) = sin(2\pi f_u t)$ avec $f_u t$ = 5 Hz,
$v(t) = cos(2\pi f_c t + \phi_c)$ où $f_c t$ = 50 Hz.

La Figure 1.3 montre le signal modulé $s_{pm}(t)$ en phase pour une phase initale ddu carrier signal ϕ_c = 0 avec un index de modulation de phase K_p = 8.

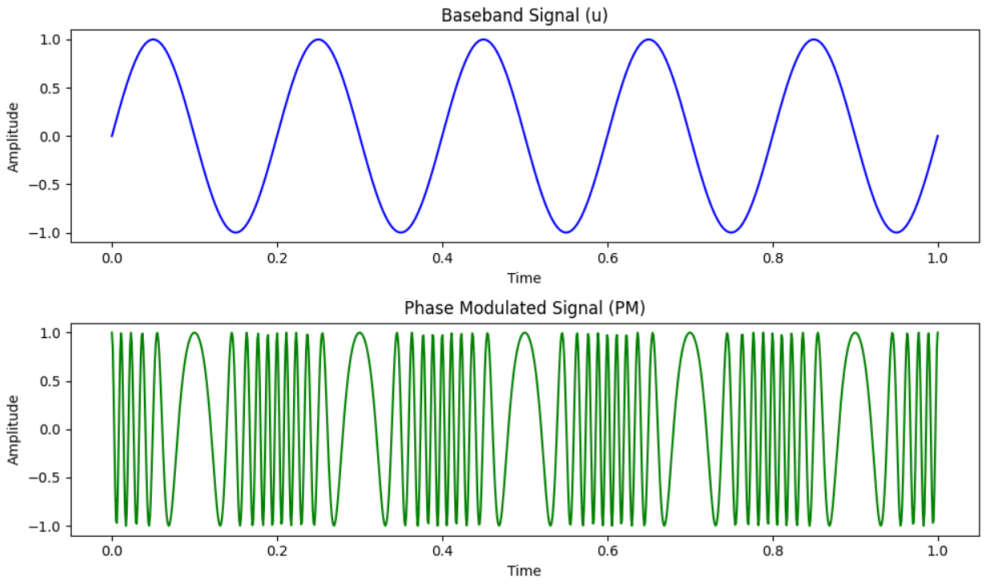

FIGURE 1.3 – Exemple de modulation en phase

1.2.2 Gestion du bruit

L'un des attributs cités concerne le bruit. Un signal est toujours affecté de petites fluctuations plus ou moins importantes, et dont les origines peuvent être diverses. Ces perturbations, appelée bruit ou *noise* en télécommunication se définissent par l'altération non souhaitée de l'intégrité d'un signal. Le bruit peut prendre différentes formes, des perturbations essentiellement impulsionnelles engendrées par des commutations de courants ou alors du bruit de fond généré dans les câbles et les composants électroniques en raison des mécanismes statistiques de la conduction électrique. Il est possible de réduire voir éliminer l'influence des perturbations impulsionelles. En revanche, le bruit de fond est lui irreductible. Tout signal sans bruit n'existe pas, même à l'émission. Il est cependant possible que le bruit devienne invisible si son niveau est très faible. L'attribut SNR est donc un critère de la qualité du signal.

1.2.3 Transformée de Fourier

Pour effectuer une analyse de signal, sa représentation est capitale. Les Figure 1 et 2 représentent des signaux en fonction du temps écoulé. Il est possible de représenter des signaux selon une autre composante, la fréquence.

La transformée de Fourier est un outil fondamental utilisé pour analyser et décomposer des signaux complexes en composantes fréquentielles. En trans-

formant un signal dans le domaine temporel en sa représentation dans le domaine fréquentiel, la transformée de Fourier révèle les différentes composantes fréquentielles présentes dans le signal. En fonction du type de signal, la transformée de Fourier est adaptée.

Pour les signaux continus, la *CFT* (Transformée de Fourier continue) convertit une fonction du temps en fonction de la fréquence en intégrant le signal par rapport aux sinusoïdes de toutes les fréquences possibles. Cette transformation fournit les informations d'amplitude et de phase pour chaque composante de fréquence présente dans le signal.

Pour les signaux discrets et échantillonnés, la *DFT* (Transformée de Fourier discrète) calcule un ensemble fini de composantes de fréquence. Il est calculé à l'aide d'un nombre fini d'échantillons, ce qui donne des composantes de fréquence discrètes. Il existe un méthode simplifiée pour les signaux discrets appelé *FFT (Fast Fourier Transform)*[11]. Il s'agit d'un moyen plus rapide de calculer la transformée de Fourier, en particulier pour les signaux numériques comportant un grand nombre de points de données. L'avantage principal de cet algorithme permet de réduire le temps de calcul en divisant la DFT en sous problèmes. La FFT est une méthode très utilisée pour l'analyse de signaux.

1.3 LoRa

LoRa (Long Range) est une technologie de communication sans fil qui permet de transmettre des données sur de longues distances avec une faible consommation d'énergie. Elle a été développée par la société française Cycleo et est maintenant gérée par la fondation LoRa Alliance, qui regroupe plusieurs entreprises et organisations du monde entier.

LoRa est principalement utilisée dans l'IoT. Elle se distingue par sa portée étendue, qui peut atteindre plusieurs kilomètres en milieu urbain et plusieurs dizaines de kilomètres en milieu rural, ainsi que par sa faible consommation d'énergie, qui permet de prolonger la durée de vie des appareils connectés. Une longue portée avec un puissance limitée induit une plus faible bande passante que les autres technologies sans fil (le Wifi, la 4G, Bluetooth etc).

LoRa utilise une bande de fréquences qui varie selon les régions du monde où LoRa est déployée :

- en Europe, la bande de fréquences autorisée est comprise entre 863 et 870 MHz, ce qui correspond à l'*ISM radio band*, un bande dédié aux recherches qui ne nécessite pas de license d'émission.
- aux États-Unis, elle se situe entre 902 et 928 MHz,
- en Chine, la fréquence autorisée varie entre 779 et 787 MHz,
- les régions restantes ont elles aussi une fourchette unique.

La technologie LoRa utilise la modulation appelé *Chirp Spread Spectrum Modulation* (CSS). La modulation CSS utilise un signal chirp, c'est à dire un signal modulé en fréquence linéaire. Ce signal a une amplitude constante mais balaie tout le spectre de la bande passante de manière linéaire dans une période de temps définie. Cette technique de modulation sera détaillée à la section 1.3.1.2

La technologie LoRa utilise également une technique de multiplexage en temps partagé (*Time Division Multiple Access*) pour permettre à plusieurs appareils de partager la même bande de fréquences de manière à maximiser l'utilisation de la capacité de transmission. Elle utilise également une technique de diffusion de données (*multicast*) pour envoyer les mêmes données à plusieurs appareils simultanément, ce qui permet de réaliser des économies de bande passante et d'énergie (source : LoRa Alliance)

En plus de sa portée étendue et de sa faible consommation d'énergie, LoRa se distingue par sa sécurité de transmission, qui est assurée grâce à l'utilisation de codes de sécurité uniques et à la possibilité de chiffrer les données transmises. Lora n'est pas exclusivement lié au protocole LoraWan. Ce protocol sera décrit en détails à la section 1.3.2. Si LoRa opère à un niveau plus bas que la plupart des protocoles réseau, LoRaWan via son infrastructure (notament les *gateway*

permet entre autre aux appareils LoRa de pouvoir utiliser différents protocoles et d'être compatibles avec un grand nombre de protocoles de communications comme *TCP/IP (transport layer protocol), HTTP (hypertext transfer protocol ou MQTT(message queuing telmetry transport).*

Toutes ces particularités font de LoRa une technologie complémentaire à celles déja existantes plutot que rivale. LoRa se compose de deux éléments principaux : la couche physique de la technologie et LoRaWAN, la couche MAC (*Media Access Control*, une sous couche de la couche liaison de données dans le modèle OSI *Open Systems Interconnection*. la couche physique de LoRa gère la fréquence radio ainsi que la modulation. LoRaWAN gère les aspects réseaux comme la sécurité, la propagation, l'adressage et la sécurité.

1.3.1 couche physique LoRa

1.3.1.1 Découpage de la couche physique

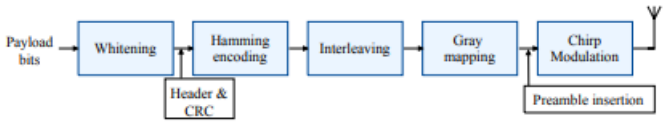

FIGURE 1.4 – Etapes de la transformation des données dans un émetteur LoRa[22]

Les étapes de la conception de l'envoi de données dans la couche physique de LoRa sont montrés dans la figure 1.4 :

— Le codage de canal (*channel coding*) est une technique utilisée dans les systèmes de communication sans fil pour améliorer la robustesse et la fiabilité de la transmission des données. Dans le cas de LoRa, le codage de canal utilise le *Forward Error Correction (FEC)* pour corriger les erreurs causées par du bruit. La méthode FEC ajoute de l'information redondante sur les données.

— Le mélange de canal (*channel interleaving*) suit le codage de canal. Cette technique consiste à réarranger les bits ou les symboles de données en les dispersant sur plusieurs canaux (ici on fait référence à des streams ou a des bandes de fréquences spécifiques plutôt qu'à des canaux physiques). Cela permet de réduire l'impact de *burst errors*, des erreurs consécutives.

— Le blanchiment de canal (*channel withening)* est la dernière étape avant la modulation du signal. Cette technique consiste à utiliser une transformation aléatoire ou pseudo-aléatoire des données avant de les trans-

mettre, de manière à répartir le spectre des fréquences de la transmission sur une large gamme de fréquences. C'est une technique mathématique qui consiste a effectuer une transformation linéaire des données avec un matrice de covariance en un nouveau set de données dans la covariance est la matrice d'identitée. Le du blanchiment est de réduire la corrélation entre les différentes composante fréquencielles et assurer que le signal possède une puissance similaire tout le long de son spectre.

— La modulation CSS est l'étape principale de LoRa. En effet, les étapes précédentes sont communes à de nombreuses technologies, mais la particularité de LoRa provient de la modulation. Cette étape est détaillée dans la section 1.3.1.2.

Chacune des étapes décrites doit être inversément réalisée pour le récepteur. Ainsi pour la récupération de donnée à l'arrivée, l'appareil récepteur gère la démodulation, le déblanchiment, le démellement et de décodage.

Cette analyse a été faite en *Reverse Engeneering* par Alexandre Marquet, Nicolas Montavont et Georgios Z. Papadopoulos [15]. Le reverse engineering consiste à analyser un produit ou un système afin de comprendre comment il fonctionne ou d'identifier ses principes de conception. Dans le contexte de LoRa, le reverse engineering examine la technologie derrière LoRa afin de comprendre ses principes de base et sa conception.

1.3.1.2 Modulation CSS

Contrairement aux modulation classique en amplitude ou en fréquence, la modulation CSS étale le signal sur une large bande de fréquence. La modulation en fréquence est linéaire et utilise des chirps. un chirp est un signal dont la fréquence change en continue tout en conservant une amlitude constante. Il existe deux types de chirps : les $upchirp$ et $downchirp$. Dans un upchirp la fréquence augmente avec le temps tandis que dans un downchirp la fréquence diminue. Soit $s_{chirp}(t)$ un signal chirp avec

$$s_{chirp}(t) = sin(2\pi(f_0 + (\frac{f_1 - f_0}{T})t)t) \tag{1.9}$$

alors la figure 1.5 montre s_{chirp} en fonction du temps où f_0 = 10Hz, f_1 = 100Hz et T = 1 seconde. On observe que le signal oscille de plus en vite plus vite au fur et à mesure que le temps augmente.

Le signal est donc séparé sur une large bande de fréquence, permettant par exemple plusieurs transmissions sans causer d'interférence. La modulation CSS est l'une des principales contributions au fait que LoRa possède une faible consommation et une longue portée. Cette technioque est très bien intégrée aux appareils a faible puissance utilisé par LoRa.

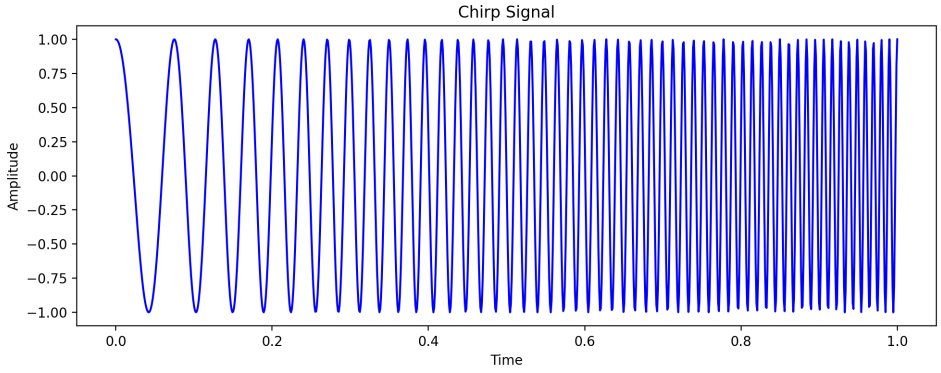

FIGURE 1.5 – Example d'un signal modulé en upchirp en fonction du temps

1.3.1.3 Spreading factor

LoRa permet d'envoyer des paquets sur une longue distance à faible puissance. Selon l'environement dans lequel les appareils LoRa sont présents, il peut être utile de pouvoir ajuster certaines capacités.

Le facteur d'étalement (*Spreading Factor, SF*) permet de déterminer le taux de variation de fréquence pour un signal. Modifier le spreading factor ajuste différentes propriétés de la communications [1]. Par exemple, si on augmente le spreading factor, les quatre conséquences principales sont :

— l'augmentation de la portée. En effet augmenter le SF réduit le bitrate et augmente le *processing gain* (l'augmentation de la puissance du signal atteint en l'étalant sur une plus large bande).
— Augmentation de la résistance aux interférences. Comme le signal est étalé sur une bande plus largeur, il y a moins de risque de subit des interférences.
— Plus petit débit de données. Le spreading factor controle le taux de chirp, et du coup la vitesse de transmission de donnée. Augmenter le speading factor signifie ralentir la vitesse d'émission des chirps. Pour chaque augmentation du spreading factor, le taux de transmission de donnée est réduit de moitié.
— Plus faible consommation. Les données transmises à un taux plus faible consomment moins d'énergie, ce qui prolonge la durée de vie des appareils dont l'économie d'énergie est une priorité.

Diminuer le spreading factor engendre l'effet inverse [1].

1.3.1.4 Structure d'un paquet LoRa

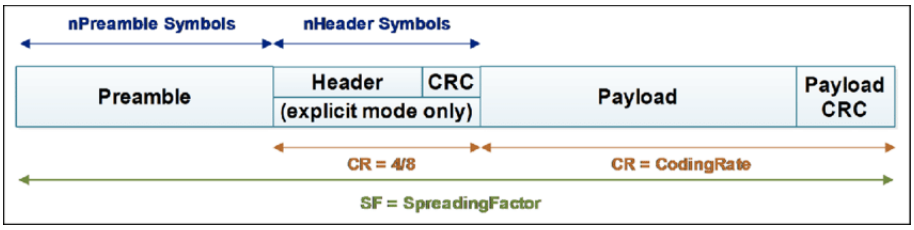

FIGURE 1.6 – Structure d'un paquet Lora[10]

La figure 1.6 montre la structure d'un paquet Lora. Un paquet LoRa contient 3 parties différentes [22] :

— Le *preamble*. La première partie du paquet, composée d'un nombre variable d'upchirps. La valeur par défaut est fixée à 8 upchirps minimum. L'émetteur radio ajoute à cela un peu plus de 4 symboles (4.25), qui contiennet l'identificateur réseau ainsi que deux downchirps de synchronisation de fréquence. Ceci fixe le préambule à 12 symboles.
— Le *header* du paquet. Les informations sur la taille du paquet, le code rate, la présence d'un CRC (*cyclique redundancy check*) et la checksum sont incluses dans l'en-tête.
— Le *payload*. La dernière partie du paquet qui contient les données à transmettre. La taille maximale du payload est de 255 octets. En plus des données, le payload peut également contenir un CRC pour la détection d'erreurs. La longueur du CRC est généralement de 16 bits.

1.3.2 LoRaWAN

LoRaWAN est un protocol de type *Low Power Wide Area Network (LPWAN)* désigné pour la communication longue portée. Ce protocole opère avec la technologie LoRa et lui fournit une infrastructure capable de maintenir une communication à longue portée et à faible cout dans l'IoT.

1.3.2.1 Aspets généraux de la technologie

LoraWan bénéficie donc d'une faible puissance de consommation et d'une portée accrue. Elle est également efficace dans différents environements. Le signal est capable de pénétrer diverses terrains et structures. Le déploiment d'une infrastructure LoRa ne nécessite pas de license, et son réseau peut être public ou privé. Les caractéristiques générales de LoraWan sont disponibles sur le site The Thing Network.

Le coeur de LoRaWAN réside sur la gestion de l'énergie, permettant aux appareils de fonctionner avec une consommation d'énergie minimale, prolongeant leur durée de vie tout en garantissant une fonctionnalité à long terme. A cette caractéristique de faible consommation d'énergie s'ajoute ses capacités en termes de portée, capable de pénéter diverses environements. Cela rend la technologie efficace aussi bien milieu rurale qu'urbain. LoRaWAN opère sur une bande de fréquence qui ne nécessite pas de license d'émition, par exemple sur la bande ISM pour *Industrial, Scientific, and Medical*. Les bandes ISM, (868 MHz en Europe ou 915 MHz aux USA) sont disponibles pour l'utilisation de différentes technologies, incluant LoraWAN.

LoRaWAN possède des capacités de géolocalisation, permettant au réseau de détecter et de localiser précisément les appareils au sein de son domaine. LoraWAn utilise différentes méthodes pour localiser ses appareils comme *Received signal strengh indication (RSSI)*, *Time difference on Arrival (TDOA)*, une triangulation ou alors une combinaison de plusieurs des méthodes. Certaines de ses méthodes seront détaillées dans la section **??**

LoRaWAN utilise des protocoles de sécurité *end-to-end*, aussi bien dans un réseau public intégré que dans un réseau privé.L'architecture LoraWan (décrite en détails dans la section 1.3.2.2) contient plusieurs couches de sécurité. Au niveau des *end devices*, une routine d'identification est imposée avant l'accès au réseau. seul les appareils de confiance son donc autorisé à communiquer. Ensuite, une fois la communication commencée, les données sont chiffrées avant d'être transmise dans le réseau. Le framework sécuritaire de Lora ne se limite pas à l'autentification et au chiffrage. LoraWan gère également les mise à jour en continue par les airs, ainsi qu'une supervision continue sur d'éventuelles intrusions.

Avec toutes ces caractéristiques, LoraWan s'est développé dans de nombreux domaines aussi bien environementaux qu'industriels. Les principales utilisations de LoraWan actuelles sont les suivantes (toutes les applications ici) :

- la surveillance environementale en général [16]. Lorawan peut être déployé pour surveiller des niveaux de températures, d'humidité, de bruits ou encore d'autres paramètres dans n'importe quel milieu. Une compagnie Hollandaise, Sensoterra, utilise notamment LoraWan pour surveiller la qualité des sols.
- Les *smart cities* [5]. LoraWan est actif sur différents aspects comme la gestion intelligent de l'éclairage, la gestion des déchets, la surveillance, etc.
- l'embarqué industriel [4]. La maintenance et la surveillance de matériel et de l'équipement peut être gérée par Lorawan. TataSteel, une compagnie indienne, utilise LoraWan pour ces équipements industriels.
- la prévention de catastrophe naturelle. Que ce soit en prévision[25] ou

après[3] d'éventuelles catastrophes naturelles, la longue portée et la surveillance en temps réel sont des atoux cruciaux pour ce genre d'évènement.

Cependant, toutes ses caractéristiques entrainent un certain nombre de limitations. La restriction de la fréquence en fonction de la région peut rendre le déploiment d'une même infrastructure à différent endroit dans le monde plus difficile. Cela peut aussi entrainer des problèmes de compatibilité entre régions, notament pour des chaines logistique ou d'approvisionement qui en traverse plusieurs.

Une faible consommation de puissance avec une grande portée a un impact sur la taille et la vitesse de l'information. La taille du payload d'un message est limitée entre 51 et 241 octets. La vitesse de transmission est également peu élevée, atteignant un maximum de 5.5kbps sur une largeur de bande de 125kHz.

La communication au sein d'un réseau LoraWan se fait en grande partie de manière asynchrone. La synchronisation dépend de la classe de l'appareils, qui est détaillé dans la section 1.3.2.2. C'est un avantage pour maintenir une grande autonomie de batterie pour les appareils. LoraWan possède un système pour limiter les colisions entre messages si plusieurs appareils communiquent simultanénent. Ce système est basé sur une combinaison entre *Listen before talk (LBT)* et des delais aléatoires[12]. Il est néamoins possible que dans un environement très dense des collisions puissent encore se produire. La comunications asynchrone et le système d'évitement de collision entraine une augmentation du temps entre les envois et la réception de message.

1.3.2.2 Topologie de Lorawan

La figure 1.7 montre les 4 types d'appareils qui composent la topologie d'un infrastrucure LoraWan. Les end devices sont les noeuds qui collectent les informations à envoyer à travers le réseau. Ils sont catégorisés en trois sous classes : A, B et C. Les apareils de classe A sont les plus économes en énergie. Ils ont été créés pour conserver leur énergie et communiquent exclusivement en comunication asynchrone. Les appareils de classe A écoutent les messages provenant des serveurs uniquement après avoir eux-même transmis un message. la classe A regroupe les appareils les moins énergivores. Les appareils de classe B sont assez similaires à ceux de classe A, mais sont occasionellement synchronisés avec les serveurs du réseau. Ils possèdent des capacités supérieurs de reception leur permettant de se synchroniser avec le scheduler des serveurs, ce qui augmente considérablement l'efficacité du temps de réponses dans le réseau. Finalement, les appareils de classe C sont en écoute permanente de messages provenant des serveurs. Ils sont les plus réactifs mais également les plus énergivores. Les end devices sont donc classés selon deux paramètres : leur réactivité et leur consomation

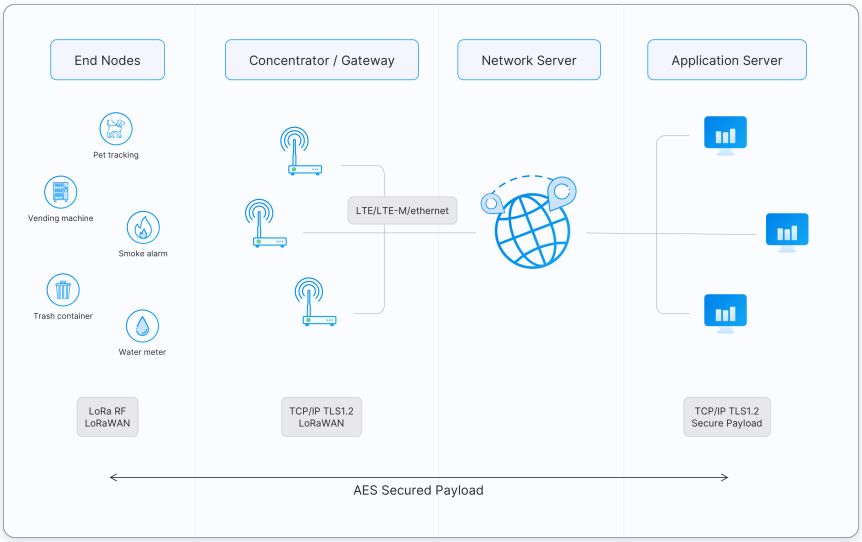

FIGURE 1.7 – Topologie de l'infrastructure LoraWan (source : The Thing Newtork)

d'énergie. En fonction de leur classe, ils ont également la possibilité de recevoir des messages server après avoir transmis de l'information. L'envoie d'un message d'un end devices vers les serveurs est appelé *uplink message* et l'envoie d'un message depuis les serveur vers les end devices est appelée *downlink message*.

Les gateways jouent le role d'intermédiaire entre les end devices et le serveur réseau. Ils recoivent les transmissions depuis les end devices dans leur zone de couverture et forward les messages vers le serveur réseau. Les gateways peuvent écouter plusieurs fréquence simultanénent (*multichanneling*) là où les end devices n'écoutent qu'une seule fréquence. Les gateway gèrent la communication radio avec les end devices en utilisation la modulation de LoRa.

Le serveur réseau est la composante centrale de l'infrastructure. Il gère tout le réseau, que ce soit les données reçues des gateays, l'identification et l'activation des end devices dans le réseau, le routing ou encore l'adapation du data rate. Le serveur réseau supervise également l'aspect sécurité au sein du réseau en gèrant les clés de chiffrage et les protocoles de sécurité.

Le serveur application de LoraWan reçoit les données forwardées depuis le serveur réseau. C'est l'interface entre le réseau de LoraWan et les différents services ou applications d'utilisaturs finaux. Les utilisateurs intéragissent avec le serveur d'application pour n'importe quelle action a effectuer sur le réseau ou pour la récupération de données du réseau. Les données reçues par le serveur réseau sont traduites par le server d'application avant d'être interprétée par l'utilisateur

final.

1.3.2.3 Sécurité

La sécurité dans l'architecture LoraWan se concentre sur trois axe principaux :

- l'authentification : qui communique avec qui.
- L'intégrité : les données ne sont pas altérées entre l'émetteur et le recepteur.
- La confidentialité : les données ne sont visible par personne au sein du réseau hormi l'émetteur et le récepteur.

La sécurité repose sur le chiffrage des données. Les données sont chiffrées en utilisant l'algorithme de cryptographie AES (*Advanced Encryption Standard*). La taille des clés est de 128 bits. Ce choix est motivé par un équilibre entre une sécurité suffisante et une consommation réduite des ressources[23].

Il y a deux types de clés utilisées dans LoraWan. La *root key* est la clé partagée entre un end device et le serveur réseau. Cette clé est utilisée pour l'authentification initiale et l'établissement d'une communication entre deux éléments du réseau. Cette clé n'est jamais transmise par les air, elle est stockée dans un *join server*. Un join server est un server dédié au contenu sensible à l'activation du matériel dans un réseau LoRaWAN. Il autentifie le réseau et les application du servers. Il gère les root keys. il génère également le second type de clés de LoraWan, les *session keys*.

Les session keys sont des clés générées dynamiquement par le join server et utilisées durant l'échange de données pendant une session. Il y a deux session keys différente, la *AppSKEY* pour le chiffrage des payloads d'application, et la *nwkSKEY* pour les fonctionalités du réseau (le chifrage à la couche MAC, les vérification d'integrité, etc).

1.3.2.4 Session

L'établissement d'une session entre un end devices et le réseau LoraWan peut se faire de deux façons différentes.

La première méthode est une méthode dynamique appelée *Over the Air Activation(OTAA)* et se déroule de la façon suivante :

- Le device possède initialement deux indentificateurs, un DevEUI et un appEUI.
- La requête pour rejoindre le réseau est initiée par le end device. Il en envoir un message *join request* au serveur réseau. La join request contient ses identificateurs, ainsi qu'un nombre aléatoire généré par le device.

— Le serveur réseau accepte (ou décline) la requête et vérifie les identifiateurs du device dans ses enregistrements.
— Si la requête est acceptée, le server génère ensuite un nombre aléatoire appelé *DevNonce* et renvoie un message *join accept* contenant le DevNonce, l'adresse du device ainsi que les clés (NwkSKey et appSKey) de session.
— le end device reçoit le message join accept. Il extrait les clés envoyées et calcule ses propres clés de session avec ses paramètres (les clés envoyé par le join server ainsi que le devNonce).
— le device fait maintenant parti du réseau. Chaque message que le device va envoyer au serveur sera chiffré avec ses clés.

La seconde méthode est hardcodée et permet à un end device de rejoindre directement le réseau sans passer par l'indentification. cette méthode est appelée *activation by personalisation (ABP)*. Voici la procédure de la session :

— Le device possède à l'avance son adresse ainsi que ses clés de session.
— Le device est déploiyé au préalable dans la zone de couverture du réseau LoraWan.
— Sans devoir itinialiser de procedure *join request*, le end device transmet directement ses données au serveur en utilisant ses clés préconfigurées. L'échange de clés avec le serveur n'a pas lieu.

Cette seconde procèdure a comme avantage d'être plus rapide à exécuter car toute la partie d'initialisation est passée. Le processus d'initialisation peut être contraignant en ressource ce qui rend la méthode ABP moins énergivore. Cependant l'utilisation de clés hardcodées directement dans les devices est une pratique moins sécuritaire. Comme pour la taille de clés, il y a un équilibre entre consommation d'énergie et sécurité.

Chapitre 2

Travaux similaires et autres contributions

Ce chapitre a pour but d'établir un état de l'art dans le domaine de l'internet of things. Comme l'iot est très vaste, seulement certains aspects seront présentés. Tout d'abord, il est important de comprendre que la sécurité dans ce domaine a évoluée autan bien par ces méthodes que par son importance, ainsi un historique est présenté dans la section2.1.1 de ce chapitre. Ensuite, la seconde section se concentre uniquement sur la technologie Lora. Cette section donne une approche analytique des signaux. Finalement, la dernière partie de ce chapitre se consacre aux approches existantes pour l'identification d'appareils utilisant Lora. Cette section présentera les travaux qui sont au centre des expérimentations de ce travail.

2.1 Identification d'appareils dans l'iot

Avante de s'intéresser à l'identification d'appareils Lora, il est nécessaire d'étendre ce concept à l'iot. L'internet of things a connu une forte évolution depuis les début des années 2000, avec des priorités dans son évolution qui ont également changées au fil du temps.

2.1.1 historique et évolution des préoccupation de sécurité dans l'iot

Afin de bien comprendre les enjeux des différentes époques, voici une petit historique des deux décénnies précedantes. Les technologie mais également le concept même d'internet of thing ont évolué.

Les premières années de l'iot (*début des années 2000*).

Bien que le concept d'appareils connectés remonte aux années 70, l'avènement de l'internet of thing arrive en fin de milénaire. Ce concept est associé à la technologie RFID *Radio Frequency Identification*[26], qui permet d'utiliser les ondes radios afin d'identifier des objets ou des personnes. Le but initial était de rendre tout objet dans le monde identifiable par un code EPC ou *Electronic Product Code*[24], un peu comme le code barre. Durant ses années, plusieurs entreprises lancent leur première appareils connectés. Tout cet enthousiame pour la connectivité met au second plan les questions de sécurité. Ainsi la première partie du développemnt de l'iot se concentre surtout sur la qualité de communication entre les objets plutot que sur leur sécurité.

Les premières préoccupations sécuritaires (*fin des années 2000, début des années 2010*)

Vers la fin des années 2000, l'augmentation du nombre d'appareils est si grande qu'elle a atteint tous les domaines de la société. Certains domaines étant plus critique que d'autre d'un point de vue sécurité (l'énergie, les transports, la santé, etc),l'intégrité des données, la confidentialité et les accès réseaux deviennent le centre de l'attetion. Le concept de certificats x.509, initalement développé pour le world wide web avant les années 2000, a un regain d'attention dans cette période. Plus largement, la structure de la technologie PKI (*Public Key infrastructure*, qui utilise les certificats x.509)[6] a été adaptée pour s'intégrer aux problématiques de l'embarqué. Un certificat est un document digital permettent de vérifier d'identité d'une entité, comme d'un appareil, un utilisteur ou une organisation. Il se base sur la liaison d'une clé public à l'entité établie par une *Certificate Authority (CA)*. La CA agit en temps de tier de confiance et assure la légitimité de l'information grâce au certificat. Ainsi, les trois axes principaux de la sécurité dans l'IoT émergent : l'autentification, l'intégrité des données et la confidentialité.

L'avènement du *Edge Computing*(à partir des années 2010).

Le nombre d'appareils connecté dépasse le nombre d'êtres humains, forçant une transition vers l'*ipv6* tant le nombre d'appareils est évelé et continue d'augmenter. L'information a pris de la valeur, et de l'ampleur. Ainsi, viens se gréffer de nouveau enjeux économiques en plus des enjeux sécuritaires. La quantité de données générées nécessite de revoir le stockage de l'information. C'est ainsi que va apparaitre le Edge computing[18], qui est un réponse directe aux besoin des architectures de gérer autan de données en périphérie de réseau. Le concept du edge computing vise à effectuer des calculs et des analyses des données directement sur les appareils connectés, plutôt que de les envoyer vers un centre de données centralisé. Cela réduit la latence, améliore l'efficacité du réseau et permet des analyses en temps réel. Le premier malware spécialemnt centré sur l'iot fait son apparition. Mirai[19] exploite une faille lui permettant de récupérer le mots de passe d'appareils afin de s'en servir pour lancer des attacks *DDoS (distri-*

buted denial of service) à grande échelle. En quelques années l'IoT est passé d'un gadget d'entreprise à un vériable enjeu économique et sécuritaire, centré autour de l'information. Les seules perspectives de législation concernant la sécurité de l'internet of thing n'apparaitront de tradivement en fin de décennies avec La loi européenne sur le *Réglément générale sur la protection des données*(source : loi RGPD. cette loi ne couvre pas la sécurité des appareils mais plutot l'utilisation des données sur internet en général.

L'apparition de la *Blockchain*(fin année 2010).

Le stockage et la transmission de données, l'authentification d'appareils ou encore la confidentialité sont au centre des préoccupations. Initialement utilisée dans les cryptomonnaie, la technologie Blockchain est un mécanisme de base de données qui permet un partage transparent des informations au sein d'un réseau. Une base de données Blockchain stocke les données dans des blocs qui sont reliés entre eux dans une chaîne. Les données sont chronologiquement cohérentes, car il n'est pas possible de supprimer ou modifier la chaîne sans le consensus du réseau. Par conséquent,la technologie Blockchain peur servir de livre inaltérable ou immuable pour le suivi des ordres, des paiements, des comptes et d'autres transactions. Le système dispose de mécanismes intégrés qui empêchent les entrées de transactions non autorisées et créent une cohérence dans la vue partagée de ces transactions. L'implémentation de la blockchain pour l'iot confère les avantages suivant[8] :

- l'immuabilité. la blockchain permet de créer un enregistrement immuable de toutes les interactions et communications des appareils. Cet enregistrement peut être utilisé pour détecter et empêcher l'accès non autorisé ou la modification des appareils ou des données dans l'IoT.
- La décentralisation. il est possible de créer un système décentralisé pour l'authentification et la communication des appareils. Chaque appareil IoT se connecte au réseau blockchain et se voir attribuer une identité numérique unique, qui est vérifiée grâce à l'utilisation de signatures numériques ou de contrats intelligents. Cela élimine le besoin d'une autorité centrale pour authentifier les appareils et annule ainsi les risques de *single point of failure*.
- La confidentialité.La technologie Blockchain peut sécuriser la communication entre les appareils IoT grâce à l'utilisation de la cryptographie à clé publique ou asymétrique. Cela permet l'échange sécurisé d'informations entre appareils sans avoir recours à des intermédiaires.

Le *Zero Trust Model*[14](début des années 2020).

les menaces de sécurité sont de plus en plus sophistiquées. Comment faire encore confiance aux infrastructures qui doivent gérer autan d'appareils ? La réponse est de ne plus leur faire confiance. Le zero Trust model est donc un modèle basé

sur l'absence totale de confiance et une vérification constante, que la demande d'accès provienne de l'intérieur ou de l'extérieur du réseau. Dans un modèle de sécurité classique, une fois qu'un utilisateur ou un appareil accède au réseau interne, on lui fait souvent implicitement confiance, ce qui lui permet une grande liberté d'actions au sein du réseau. Le Zero Trust model suppose cependant que des menaces peuvent exister à la fois à l'intérieur et à l'extérieur du périmètre du réseau et nécessite donc une vérification continue de la confiance. Un modèle qui s'applique sur ce principe devrait contenir les éléments suivants[17] :

- La vérification d'identité. Les utilisateur et les appareils doivent subir une authentification avant d'accèder à n'immporte quel services ou ressources du réseau.
- Le *Least privilege acces*. les permissions sont accordées de manières limités selon le besoin de l'utilisateur ou de l'appareil.
- La micro segmentation. Diviser le réseau en segments pour limiter son accès par les appareils.
- La surveillance en continue. L'analyse du traffic, du comportement et des activités des appareils.
- le chiffrage des données.

Le *quantum computing*(dans les prochaines années...).

Avec la future arrivée des ordniateurs quantiques, les méchanismes de chiffrage basés sur la comlexité mathématique comme RSA ou ECC sont voués à disparaitre[21]. La puissance de calcul des ordinateurs quantique est déja considérée comme une véritable menace pour la sécurité informatique. Fort heureusement, c'est également un nouveau champ de possibilité qui s'ouvre pour la sécurité, avec le développement du *post quantum cryptography*. Un premier protocol résistant au menances quantiques, *Quantum Key Distribution* permet d'établir des canaux de communications entre différents appareils dand l'iot. Ce protol n'est pas encore en service dans l'iot, mais les premiers est réalisé en laboratoire sont très prometteurs[7].

2.1.2 Approches d'identification dans l'iot

Comme tout appareil au sein d'un réseau, un appareil connecté dans l'iot possède de base différents moyens d'authentification. Son adresse MAC, son adresse IP, des informations relatives à sa manufacturation comme un numéro de série par example. De manière plus spécifiques, les noeuds au sein d'un réseau LoraWAN possède un DevEUI, un numéro unique accordé par le réseau à l'appareil. Cependant ses informations peuvent être compromises si appareils sont victimes de *devices spoofing*, c'est à dire qu'un appareil malveillant usurpe l'indentité de sa cible, afin d'accéder au sein du réseau. D'autres attaques comme *Man in The*

Middle ou *Replay attacks* peuvent également compromettre l'identité d'un appareil si on se base uniquement sur ses identifiant classiques[9]. Il faut donc pouvoir identifier les appareils mais sans se fier à leur informations. Il éxiste diverse méthodes basée sur différentes approches pour pouvoir indentifier un appareil.

La première approche possible est de fier non pas à l'appareil directement ni aux données qu'il reçoit car celles ci pourraient également être compromises, mais à la routine sur sa communication. Cette approche, appelée *Traffic related pattern* où s'intéresse au comportement d'un appareil au sein d'un réseau. L'article publié par H. Kawai, S. Ata et N. Nakamura [13] propose notament d'analyser via machine learning le *traffic pattern* c'est à dire le comportement du traffic.

Une autre approche s'intéresse à la position géographique d'un appareil. En effet il est possible de mesure la qualité de la réception d'un signal, le *Received Signal strengh Indicator* ou RSSI. Un étude a été menée sur des appareils lora par M. Anjum, MA. Khan et SA. Hassan[2]. L'article utilise le *Path Loss* pour estimer le RSSI. Via machine learning ils sont capables de recréer un système de positionement permettant l'identification d'appareil Lora.

La dernière approche se concentre sur l'analyse des charactéristiques uniques des fréquences radios. Plutot que de s'intéresser au routine entre les communication ou la distance d'où elles ont lieu, le *Radio Frequency Finguerprinting* se concentre sur les propriétés physiques des signaux. Différentes techniques sont montrées par N. Soltanieh, Y. Norouzi et Y. Yang[20].

FIN DE LA REDACTION

2.2 Analyse de la technologie lora

2.3 identification de device lora

radio frequency finguerprinting identification (RFFI). trouver des charactéristiques hardware pour identifier des devices.

2.3.1 RFFI avec DCTFs

L'analyse suivante est basé sur l'article publié par Yu Jiang, Linning Peng, Aiqun Hu, Sheng Wang, Yi Huang et Lu Zhang [27]. L'objectif de cette analyse est de pouvoir identifier un appareil LoRa uniquement en se fiant à l'analyse des signaux générés par ce dernier.

Cette analyse est séparée en plusieurs étapes.

aquisition des données. On s'intéresse uniquement au préambule du singal car cette partie est fixe peu importe le signal envoyé. Ensuite, on extrait les échantillons IQ du preambule lora.

Preprocessing des données. Si on plot les échantillons tel quel, on n'observe un cercle mais on en apprend pas plus. A cause du CFO (carrier frequency offset), les données dévies et couvre l'information importante sur le graphe. Il faut appliquer une méthode différentielle pour recalculer les données (equation dt).

Une fois que les donnés sont recaculée, on observe via un gradient coloré une zone dense dans le cercle. On calcule le centre de la manière suivante. On calcule le ploint le plus dense ainsi que les points au moins 90 pourcent aussi dense. Ensuite, on calcul le centre euclidien entre tout les point éligibles. On répète cette opération pour chaque échantillons.

Une fois qu'on a obtenu le centre pour chaque échantillons, on plot pour chaque module le cluster de point, permettant ainsi d'utiliser une méthode classification. La méthode kmeans convient pour cette nalyse car on connait à l'avance la valeur de k (le nombre de module).

2.3.2 RFFI avec spectrogrammes

article sur les spectogrammes.

Chapitre 3

Expérimentations

3.1 Matériel

3.1.1 radio logicielle

La radio logicielle (SDR, pour $Software\text{-}DefinedRadio$) est une technologie qui permet de mettre en œuvre des systèmes de radio à l'aide de logiciels plutôt que de matériel.

Dans les systèmes de radio traditionnels, les différentes fonctions de la radio, comme l'accord sur une fréquence spécifique, la modulation et la démodulation du signal, et le filtrage du bruit, sont mises en œuvre à l'aide de composants matériels tels que des oscillateurs, des amplificateurs et des filtres. En revanche, les systèmes SDR utilisent des logiciels pour effectuer ces fonctions, ce qui les rends beaucoup plus flexible car chaque composante est reconfigurable. Les radios logicielle sotn capable d'opérer sur une large portée de fréquence, aussi bien très basse fréquence comme haute fréquence. Les SDR peuvent jouer le role d'éméteur ou de récepteur voir les deux.

3.1.1.1 RTL-SDR

La première radio utilisée comme récepteur. possède différente composante :

rtl2832U : digitalise les signaux RF et les evnoie à l'ordinateur. Tuner chip : le tuner permet d'ajuster la fréquence. Grace à ça la sdr peut couvrir une larger portée. port usb : pour raccorder la sdr à l'ordinateur.

3.1.1.2 hackRf

autre radio logicielle. plus (cher et) complète. meilleure qualité de signal que les rtl-sdr "classiques".

3.1.2 Module d'émission Lora

3.1.2.1 module RN2483

Le microchip RN2483 est un module de technologie spécifique à LoRa. Cet appareil permet de communiquer à longue portée et à faible coup grêve à l'utilisation de la modulation basé sur LoRa.

quelques spécificités du module :

technologie LoRa faible puissance (ideale pour de l'iot car faible consommation) fréquence à 433, 868 et 915MHz (regarder la régions adéquate) AT command : configurable via un set de commande compatible avec le protocole LoRaWAN pour établir ou rejoindre ce type de réseau.

fonctionne par entrée de commande (aucun retour écran donc aucune faute possible)

différentes commandes/ utilisation :

sys get ver : demande la verion du module, reçoit en réponse radio set (param) (value) : ajuste le paramètre pour l'adapter a la valeur souhaitée.

3.1.2.2 pycom fipy

besoin d'un environement python , qqs configuration nécessaire.

3.1.2.3 module arduino

besoin d'un IDE arduino, possibilité de configurer une largeur de bande bien plus faible que pour les autre module, très pratique pour analyser le signal en détail.

3.1.3 logiciel

3.1.3.1 GNU radio

GNU Radio est un toolkit qui permet de créer des flux de traitement de signal en utilisant des blocs prédéfinis. Ces blocs peuvent être combinés pour créer des chaînes de traitement de signal pour simuler des modulations CSS, capturer des signaux et en extraire des séquences de chirp.

3.1.3.2 gqrx

logiciel open source d'analyse de fréquence radio pour les SDR.
installer gqrx via apt. (ubuntu)
sélectionner le périphérique pour analyse
image choix périphérique

visualisation du spectre

deux forme d'affichage,en spectre et en cascade.

L'affichage du spectre fournit une représentation graphique en temps réel du spectre RF sur une gamme de fréquences. Il montre la puissance du signal de différentes fréquences sur une plage de fréquences spécifiée. L'axe des x représente la fréquence, tandis que l'axe des y affiche la force du signal (mesurée en dB).

L'affichage en cascade est un spectrogramme qui visualise la force du signal au fil du temps. Il montre une série d'instantanés de spectre empilés les uns sur les autres, où l'intensité de la couleur représente la force du signal. Chaque ligne horizontale du tracé en cascade représente une vue du spectre capturée à un moment précis, créant ainsi un enregistrement historique de l'activité du signal. L'axe vertical représente la fréquence et l'axe horizontal représente le temps.

image affichage spectre

configuration de la réception :

imput control (pas trop touché)

FFt settings : très important règle la ff size, le raffraichissemetn d'image. le laps de temps. l'averaging

Le paramètre Panadapter dB fait référence à l'échelle verticale dans la vue du spectre. Il représente la force du signal des fréquences radio reçues affichées sur l'axe vertical du graphique du spectre. Le réglage du paramètre Panadapter dB modifie l'échelle verticale de la force du signal affichée dans la vue du spectre.

Le paramètre Waterfall dB concerne l'intensité de la couleur ou l'ombrage des fréquences affichées dans le tracé en cascade.Le réglage du paramètre Waterfall dB modifie l'intensité utilisée pour afficher la force du signal dans le tracé en cascade, permettant ainsi d'ajuster le contraste ou la visibilité des signaux plus faibles ou plus forts.

3.1.3.3 Universal radio hacker, URH

logiciel open source. similaire à gqrx pour fonction d'analyse du signal. Il est possible de visualiser les signaux de manières analogique, démodulé, en spectrogramme ou en vue I/Q. Il est possible de découper les signaux, notamment pour supprimer les parties "vides" dans les enregistrements. Possibilité de sauvegarder des signaux enregistré dans des fichier. Urh supporte différents formats pour les signaux :

.complex files with complex64 samples (32 Bit float for I and Q, respectively). This is the default signal file format.

.complex16u using two unsigned 8 Bit integers for I and Q

.complex16s using two signed 8 Bit integers for I and Q

.complex32u using two unsigned 16 Bit integers for I and Q (since v2.7)

.complex32s using two signed 16 Bit integers for I and Q (since v2.7).

Il est également possible de lire des fichiers de données qui n'ont pas été enregistré avec URH tant qu'ils sont dans les formats supportés.

3.2 Librairie python

utilisation de python : pourquoi ? librairie très utile dans le domaine comme NumPy, Pandas, Scikit-learn, PyCM et FiPy. Librairie disponible pour les radio logiciel hackrf et rtlSDR. Librairie compatible avec la sauvegarde et l'utilisation des données. Bonne documentation notament sur les formats des nombres complexes. Capacités pour le machine learning. Utilisation de diverses algorithmes comme kmeans déja implémenté dans des librairies. Intégration avec des modules, la librairie fipy gère l'un es modules utilisé pour les éxpérimentations.

numpy : pour complexe conjugué matplotlib : pour les plot des diagrammes datashader : pour la coloration du diagramme de constellations

3.3 Géneration et réception d'un signal LoRa

dans un premier temps les signaux sont généré manuellement sans automatisation, le but étant de reconnaitre et d'analyser la structure d'un signal Lora.

script pour générer un signal depuis un module (soir rn, soit fipy, soit arduino) Il faut paramètrer la largeur de bande, le CR, le spreading factor, la fréquence d'émission (fixé à 868Mhz dans la région europe), le type de modulation (Lora ...)

Ensuite pour la réception, lancer un logiciel d'écoute comme gqrx ou urh. Sélectionner la sdr connecté à l'ordinateur (rtl sdr ou hackrf). Il faut configurer également l'échantillonage, soit la quantité d'échantillons qui vont être lu chaque seconde. Il faut ajuster la fréquence d'écoute.

3.3.1 analyse du signal

affichage du signal capturé, d'abords sous forme analogique, puis en spectrogramme. Décomposition du signal, on observe des "chirps". En analogique, augmentation de la fréquence (unchirp) et diminution de la fréquence (downchirp). En spectrogramme, augmentation est plus visuelle encore, on voit de manière net les chirps.

Dans le signal, on reconnait donc le préambule du signal composé de 10 upchirps et 2 downchirps (selon la théorie).

Attention, la fréquence d'écoute des sdr ne doit pas être exactement à 868Mhz. En effet, voilà à quoi ressemble si la fréquence d'écout est la même que la fréquence d'émission. Il faut prendre en compte la largeur de bande du signal, dans le cas ou

le singal a une largeur de bande de 125KHz, il faut décaler la fréquence d'écoute pour recentrer le signal, ainsi on évite d'avoir des fréquence qui sont interprétée comme "négatives" par URH. Dans la figure, la fréquence est décalé de 125/2 soit 867,935Mhz.

3.3.2 automatisation du signal

Dans un second temps, besoin d'automatiser la génération et l'enregistrement des signaux, car il va falloir en générer un grand nombre pour les méthodes d'indentifications. La première étape consiste a éditer le script pour que la configuration des paramètres soit faite au préalable. Ensuite, sans devoir passer par un logiciel d'écoute, il faut sauvegarder le signal. La librairie RTL-SDR de python permet de configurer la rtl sdr sans devoir passer gqrx ou URH. Il est alors possible avec juste un script python d'écouter et d'enregistrer un signal radio. Il faut ajouter au script une méthode qui supprime dans la partie sauvegardée la section ne comportant pas le signal.

3.4 Méthode "Constellation traces"

objectif, identification du device via son frequecy offset.

selon l'article (citer article), possible de performer la méthode soit uniquement sur le préambule, soit sur l'intégralité du singal.

idée : le received singal contient le baseband singal ainsi qu'un rotation factor instable. pour pouvoir recover cette partie du signal, besoin d'effctuer une opération différentielle suivante :

$$x(t).x(t+n)e - j2\pi on$$

apparation d'un nouveau rotation factor, mais stable. Besoin de trouver deux inconnue,
textitdelta f et *n*. n est le differential interval. il se calcule de la manière suivante :

$$Rs = BW/2sf$$

$$N = fs/Rs$$

delta f est la difference entre le transmitter carrier frequency et le receiver carrier frequecy.

application : récupérer les samples I/Q du signal n ayant au préalable "clean" le signal. utilisation d'un gradiant pour observer la dentsité sur le plot. noramlement des zones plus denses apparaissent. coloration : utilisation de la librairie data shader.

clustering, le but de conserver les parties les plus dense (95pourcent du point le plus dense) plot avec les différents appareils. librairie panda et numpy. découpage en zones (bins ?) sous forme d'une grille, calcul de nobre de points dans chaque zone, la zone avec le plus grand nombre de point sers de maximum.

Chapitre 4

Résultats

4.1 Méthode des constellations Traces

4.1.1 paramètrage

3 modules lora
module rn2483
module pycom fipy
module arduino
paramètres d'émission : SF =7, BW = 125, Freq = 868Mhz, CR = 4/5, mod = Lora
plot des DCTF (differential constellation traces figure)

4.1.2 training phase

during the training phase, you use differential constellation trace figures (DCTFs) from different modules. The objective is to identify clusters in these figures and learn the characteristics of each module.

4.1.3 testing phase

In your case, during the testing phase, you have new DCTFs from modules that the model has not seen during training. The model uses what it learned to predict the category or cluster for each DCTF.

4.1.4 résultats

Bibliographie

[1] Lorawan spreading factors. `https://www.thethingsnetwork.org/docs/lorawan/spreading-factors/`. Accessed on 2024-02-08.

[2] M. Anjum, M. A. Khan, S. A. Hassan, A. Mahmood, and M. Gidlund. Analysis of rssi fingerprinting in lora networks. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 1178–1183. IEEE, 2019.

[3] M. I. Arsyad, A. A. Pekerti, and B. A. Nugraha. Time-slotted lorawan implementation of gps tracker system for post-disaster utilization. pages 172–178, 2020.

[4] M. Ballerini, T. Polonelli, D. Brunelli, M. Magno, and L. Benini. Nb-iot versus lorawan : An experimental evaluation for industrial applications. *IEEE Transactions on Industrial Informatics*, 16(12) :7802–7811, 2020.

[5] P. J. Basford, F. M. Bulot, M. Apetroaie-Cristea, S. J. Cox, and S. J. Ossont. Lorawan for smart city iot deployments : A long term evaluation. *Sensors*, 20(3) :648, 2020.

[6] R. Canetti, D. Shahaf, and M. Vald. Universally composable authentication and key-exchange with global pki. In *Public-Key Cryptography–PKC 2016 : 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part II 19*, pages 265–296. Springer, 2016.

[7] Y. Cao, Y. Zhao, Q. Wang, J. Zhang, S. X. Ng, and L. Hanzo. The evolution of quantum key distribution networks : On the road to the qinternet. *IEEE Communications Surveys & Tutorials*, 24(2) :839–894, 2022.

[8] D. Dasgupta, J. M. Shrein, and K. D. Gupta. A survey of blockchain from security perspective. *Journal of Banking and Financial Technology*, 3 :1–17, 2019.

[9] J. Deogirikar and A. Vidhate. Security attacks in iot : A survey. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 32–37, 2017.

[10] E. Gambi, L. Montanini, D. Pigini, G. Ciattaglia, and S. Spinsante. A home automation architecture based on lora technology and message queue telemetry transfer protocol. *International Journal of Distributed Sensor Networks*, 14 :155014771880683, 10 2018.

[11] P. Heckbert. Fourier transforms and the fast fourier transform (fft) algorithm. *Computer Graphics*, 2(1995) :15–463, 1995.

[12] Q. L. Hoang, H. P. Tran, W.-S. Jung, S. H. Hoang, and H. Oh. A slotted transmission with collision avoidance for lora networks. *Procedia Computer Science*, 177 :94–101, 2020.

[13] H. Kawai, S. Ata, N. Nakamura, and I. Oka. Identification of communication devices from analysis of traffic patterns. In *2017 13th International Conference on Network and Service Management (CNSM)*, pages 1–5. IEEE, 2017.

[14] S. Li, M. Iqbal, and N. Saxena. Future industry internet of things with zero-trust security. *Information Systems Frontiers*, pages 1–14, 2022.

[15] A. Marquet, N. Montavont, and G. Z. Papadopoulos. Towards an sdr implementation of lora : Reverse-engineering, demodulation strategies and assessment over rayleigh channel. *Computer Communications*, 153 :595–605, 2020.

[16] A. N. Rosli, R. Mohamad, Y. W. Mohamad Yusof, S. Shahbudin, and F. Y. Abdul Rahman. Implementation of mqtt and lorawan system for real-time environmental monitoring application. pages 287–291, 2020.

[17] M. Samaniego and R. Deters. Zero-trust hierarchical management in iot. pages 88–95, 2018.

[18] K. Sha, T. A. Yang, W. Wei, and S. Davari. A survey of edge computing-based designs for iot security. *Digital Communications and Networks*, 6(2) :195–202, 2020.

[19] H. Sinanović and S. Mrdovic. Analysis of mirai malicious software. In *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–5. IEEE, 2017.

[20] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar. A review of radio frequency fingerprinting techniques. *IEEE Journal of Radio Frequency Identification*, 4(3) :222–233, 2020.

[21] F. Song. A note on quantum security for post-quantum cryptography. In *International Workshop on Post-Quantum Cryptography*, pages 246–265. Springer, 2014.

[22] J. Tapparel, O. Afisiadis, P. Mayoraz, A. Balatsoukas-Stimming, and A. Burg. An open-source lora physical layer prototype on gnu radio, 02 2020.

[23] P. Thaenkaew, B. Quoitin, and A. Meddahi. Leveraging larger aes keys in lorawan : A practical evaluation of energy and time costs. *Sensors*, 23(22) :9172, 2023.

[24] F. Thiesse and F. Michahelles. An overview of epc technology. *Sensor review*, 26(2) :101–105, 2006.

[25] K.-H. Tseng, M.-Y. Chung, L.-H. Chen, and Y.-W. Huang. Implementation of composite lpwan on the slope disaster prevention monitoring system. *IEEE Sensors Journal*, 22(3) :2658–2671, 2022.

[26] R. Weinstein. Rfid : a technical overview and its application to the enterprise. *IT Professional*, 7(3) :27–33, 2005.

[27] X. Wu, Y. Jiang, and A. Hu. Lora devices identification based on differential constellation trace figure. In *Artificial Intelligence and Security : 6th International Conference, ICAIS 2020, Hohhot, China, July 17–20, 2020, Proceedings, Part I 6*, pages 658–669. Springer, 2020.

Annexe A

Première annexe

Annexe B

Deuxième annexe