HAL
open science

Réseaux de capteurs et vie privée

Jessye dos Santos

To cite this version:

Jessye dos Santos. Réseaux de capteurs et vie privée. Cryptographie et sécurité [cs.CR]. Université Grenoble Alpes, 2017. Français. NNT : 2017GREAM035 . tel-01682930

HAL Id: tel-01682930

https://theses.hal.science/tel-01682930

Submitted on 12 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES

Spécialité : **Mathématiques et Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Jessye DOS SANTOS

Thèse dirigée par **Claude CASTELLUCCIA** et
codirigée par **Cédric LAURADOUX et Christine HENNEBERT**

préparée au sein du **Commissariat à l'Energie Atomique (CEA)**
dans **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Réseaux de capteurs et vie privée

Thèse soutenue publiquement le **28 aout 2017.**
devant le jury composé de :

Monsieur Pascal Lafourcade
Maître de conférences, Université Clermont Auvergne, Rapporteur
Monsieur Benjamin Nguyen
Professeur, INSA Centre Val de Loire, Rapporteur
Monsieur Cédric Lauradoux
Chargé de recherche, INRIA Rhône Alpes, CoDirecteur de thèse
Madame Christine Hennebert
Ingénieur de recherche, CEA Grenoble, CoDirecteur de thèse
Monsieur Claude Castelluccia
Directeur de recherche, INRIA Rhône Alpes, Directeur de thèse
Madame Marine Minier
Professeur, Université de Lorraine, Président du jury

À mon pépé Poloche,
À ma tata Mumu la coquine.

Remerciements

Je voudrais remercier Pascal Lafourcade et Benjamin Nguyen pour avoir accepté d'être rapporteurs de ma thèse et pour l'honneur qu'ils m'ont fait de juger mon travail. Je remercie également Marine Minier pour son travail de présidente du jury. Je remercie Claude Castelluccia d'avoir accepté de diriger ma thèse. Je remercie mes encadrants, Cédric Lauradoux et Christine Hennebert pour leurs disponibilités, leurs appuis scientifiques et leurs conseils avisés. Vous avez su m'orienter et m'aider afin que ce travail voie le jour.

Je remercie Florian Pebay Peyroula, Bruno Charrat et Guillaume Herve, les chefs successifs que j'ai pu avoir au fil des réorganisations, de m'avoir accueilli au sein de leurs équipes.

Mon travail au sein du CEA LETI n'aurait pas été possible sans tous mes collègues qui ont su m'apporter un soutien tant professionnel que moral. Je remercie donc tous ceux qui ont, par l'élaboration de programmes, du développement d'outils ou grâce à nos discussions scientifiques, permis l'avancement et l'aboutissement de ma thèse. Ils ont su également m'écouter malgré ma tendance à me plaindre et me remonter le moral dans les moments de doute. Nos verres partagés au "Carré", le pique nique annuel (très bien organisé) mais également les CTF de Grehack m'ont permis de tenir ces trois (+1) ans. Je souhaite donc remercier tout le laboratoire LSOSP et plus particulièrement Thomas, Alexandre, Sylvain (les anciens) et Thibault.

Ma thèse m'ayant amenée à assurer des encadrements de cours, je remercie le personnel enseignant de l'école d'ingénieur Polytech Grenoble qui a su m'intégrer comme un membre à part entière. Mes plus chaleureux remerciements vont à Sylvie Charbonnier qui a pris de son temps pour encadrer mon travail d'enseignement et accepté de me conseiller sur mon orientation de carrière. J'ai toujours pu compter sur ses bons conseils.

Comme m'a dit une amie : "La pression il vaut mieux la boire que l'avoir". Alors je remercie tous ceux qui m'ont aidés à m'évader.

En premier les filles du foot. Nos entrainements pas très sérieux et les matchs du dimanche matin m'ont été bénéfiques pour décompresser. Les barbecues au rosé et les soirées au gite avant les matchs vont me manquer. Merci à Guy pour ses entrainements, sa patience et sa présence le jour de ma soutenance.

Je remercie bien sûr mes amis, ceux de longue date comme les plus récents. Merci à Aurélien, Cédric, Alexandre (malgré que tu sois parti sans me prévenir), Camille et Thibault pour nos repas et nos soirées.

Mes remerciements les plus profonds vont à ma mère Marylène, mon père Tony, mon frère Alex et ma grand mère "mémé Roro". Vous m'avez soutenu et aidé à tenir. Vous avez également été très présent lorsque j'avais besoin de vous pour me suppléer dans des tâches plus terre à terre me permettant de me consacrer, dans les moments de grandes précipitations, pleinement à mon travail. Je ne citerai pas tout le monde car j'ai la chance d'avoir une grande famille aimante mais je remercie bien sûr tous les autres membre de ma famille qui, autour des différents repas et réunions de famille, m'ont permis de décompresser. J'ai également la chance d'avoir une famille d'adoption aussi aimante. Je remercie donc ma belle famille pour leur soutien et leur intérêt pour mon travail. Merci à Rebecca, future présidente de la république et marraine de ma fille, d'avoir discuté avec moi de mon travail ainsi que de son soutien.

Enfin, je remercie mon compagnon Olivier qui, depuis 2006, m'a soutenu de manière inconditionnelle dans tous mes choix de formations du diplôme d'ingénieur à la réalisation de ma thèse. Tu as dû supporter ma mauvaise humeur, mes horaires tardifs et mes obligations professionnelles. Tu m'as permis de me changer les idées et tu as pris en charges de nombreuses tâches, mettant bien souvent tes envies de côté pour mon épanouissement. Je remercie également ma fille Lindsay née pendant ma thèse. Cela n'aura pas été tout le temps simple de concilier vie privée et vie professionnelle, j'ai souvent été absente mais tu es ma plus belle réussite.

Résumé du rapport

Les médias et de nombreuses études scientifiques évoquent fréquemment la notion de vie privée en lien avec des exemples de cyber attaques. Le vol par des *hackers* de 12 millions d'identifiants d'utilisateurs Apple en 2012 illustre que les objets communicants sont des maillons vulnérables exploités par les *hackers* pour accéder aux données personnelles des usagers. Dans cette thèse, nous allons étendre la notion de vie privée aux objets eux-mêmes, au-delà des utilisateurs, en montrant que dans des réseaux de capteurs sans fil où les communications ont lieu de machine à machine, la connaissance des adresses fixes des différents appareils constituant le réseau représente une source d'information permettant de déduire beaucoup d'éléments de contexte et d'environnement.

Actuellement, tous les standards de communication sans fil intègrent la capacité de sécuriser les données transportées, y compris les protocoles de communication dédiés aux réseaux de capteurs, conçus pour fonctionner en milieu contraint et à basse consommation. Cependant, l'en-tête des trames envoyées sur l'air comportant les informations nécessaires au routage et au bon fonctionnement du réseau, figure toujours en texte clair. La collecte de ces métadonnées par écoute passive représente un danger pour les environnements et les applications qui font usage de ces réseaux.

Le travail mené dans cette thèse a pour objectif d'explorer comment de simples attaques passives sur des réseaux meshés basés sur le standard IEEE 802.15.4, visant à collecter et exploiter les métadonnées de ces trames échangées sur l'air, permettent d'inférer des informations critiques sur le réseau lui-même, l'environnement dans lequel il est déployé et les comportements des personnes qui en font usage.
Plusieurs solutions visant à dissimuler les adresses des nœuds du réseau sont ensuite étudiées. Ces solutions sont de deux types : soit elles rendent anonymes les dispositifs empêchant de remonter à la source des messages, soit elles reposent sur l'utilisation de pseudonymes permettant de conserver la possibilité d'auditer le trafic.

Afin d'évaluer les caractéristiques et les performances de ces solutions, un simulateur a été mis en œuvre afin de reproduire le comportement d'un réseau de capteurs meshés embarquant l'OS Contiki. Ce simulateur a permis d'évaluer la solution la plus prometteuse issue de l'état de l'art, nommée MT6D, en comparant ses performances avec un réseau de référence ne dissimulant pas les métadonnées. Cette analyse a fait ressortir certains inconvénients, en particulier l'augmentation importante des trames de contrôle nécessaires au routage, et a permis d'élaborer les spécifications d'une solution plus optimale pour l'embarqué.

Nous avons ainsi introduit Ephemeral, qui présente la capacité de dissimuler les adresses des dispositifs dans les messages envoyés sur l'air, par l'usage de pseudonymes, sans augmenter la quantité de trames de contrôle indispensables au routage. Une fois mis en œuvre avec le simulateur afin de valider les performances théoriques attendues, Ephemeral est déployé en environnement réel sur un réseau de capteurs IEEE 802.15.4 équipant un bâtiment. Ce retour d'expérimentation permet de confirmer qu'Ephemeral constitue une solution économe du point de vue de la consommation d'énergie et de la bande passante du réseau, pour masquer les identifiants des dispositifs impliqués dans les communications.

Mots clés : Réseaux de capteurs sans fil, Sécurité, Protection de la vie privée, Adresses MAC, Pseudonymes, IEEE 802.15.4, 6LoWPAN, ZigBee, Simulation, Déploiement.

Abstract

Title : Wireless Sensor Networks and privacy

Privacy notion is frequently linked with cyber attack examples by media and scientific researches. In 2012, the hacking of 12 millions Apple user identifiers demonstrates that connected objects represent leaks exploited by hackers to access to user personal data. In this thesis, we will extend the privacy notion to the objects. To do this, we will show that in wireless sensor networks where communications are carried out from machine-to-machine, the knowledge of the static addresses of the devices within the network discloses information allowing deduction about elements of context and environment.

Nowadays, the wireless communication standards provide security mechanisms whatever the communication protocols used including the low power ones designed to run on constrained environment. However, the frame header that comprises necessary information for routing and for the proper functioning of the network

is always sent in clear text. Collecting and gathering these metadata by eavesdropping is dangerous for the environments and applications based on these networks.

The work carried out in this thesis aims to explore how simple passive attacks on meshed networks based on IEEE 802.15.4 used to collect and exploit metadata allow to infer critical information about the network, the environment where the network is deployed and the behavior of users.
Two kinds of solutions to hide the node addresses are studied. The first one provides anonymity for the devices. In the second kind of solutions, pseudonyms are used by nodes enabling the capability to audit the traffic within the network.

To evaluate the characteristics and the performances of the solutions, a simulator has been used to reproduce the behavior of a meshed wireless sensor network embedding Contiki OS. This simulator allows to compare the performances of MT6D the most promising solution of our state of the art with those of a reference network not hiding the metadata. With this analyze, we can highlight some drawbacks and more especially the control frames overhead needed for the routing. We give the necessary specifications to deploy the most optimal solution for the embedded devices.

Thus we propose Ephemeral that allows hiding device addresses provided in the sent frames by using pseudonyms without overhead on the control frames. After deployment in the simulation environment to evaluate expected theoretical performances, Ephemeral has been tested in real environment. The network is made up of twenty IEEE 802.15.4 sensor nodes deployed on a building. The results show that Ephemeral is an efficient low power and bandwidth-saving solution to hide device identifiers used in wireless communications.

Keywords : Wireless sensor networks, Security, Privacy, MAC addresses, Pseudonyms, IEEE 802.15.4, 6LoWPAN, ZigBee, Simulation, Deployment.

Table des matières

1 Introduction générale **21**
1.1 L'IoT . . . 21
1.2 Les réseaux de capteurs . . . 21
1.3 La sécurité des WSN : un challenge actuel . . . 23
1.4 Organisation du mémoire et contributions . . . 24

2 Les vulnérabilités dans les WSN **27**
2.1 Introduction . . . 27
2.2 Les réseaux de capteurs . . . 29
2.3 Apport de la cryptographie à la sécurité des WSN . . . 31
2.3.1 La confidentialité . . . 31
2.3.2 L'authentification du message et l'intégrité des données . . . 33
2.3.3 La distribution des clés . . . 33
2.4 Les attaques contre la sécurité dans les WSN . . . 34
2.4.1 Les modèles d'attaquant . . . 34
2.4.2 Les attaques contre la sécurité . . . 35
2.4.2.1 Attaques sur la couche Physique . . . 35
2.4.2.2 Attaques sur la couche MAC . . . 35
2.4.2.3 Attaques sur la couche Réseau . . . 36
2.4.2.4 Attaques sur la couche Application . . . 37
2.4.3 Etat de l'art des contre mesures . . . 38
2.4.3.1 Contre mesures sur la couche Physique . . . 38
2.4.3.2 Contre mesures sur la couche MAC . . . 38
2.4.3.3 Contre mesures sur la couche Réseau . . . 38
2.4.3.4 Contre mesures sur la couche Application . . . 39
2.4.3.5 Mesures de détection . . . 40
2.5 La protection de la vie privée . . . 41
2.5.1 Les modèles d'attaquant . . . 41
2.5.2 Les attaques . . . 42
2.5.3 Etat de l'art des contre mesures . . . 43
2.6 Conclusion . . . 50

3 La sécurité des protocoles radio basés sur IEEE 802.15.4 **53**
3.1 Introduction . . . 53
3.2 Positionnement du standard IEEE 802.15.4 parmi les standards radio grand public . . . 54
3.3 Le standard IEEE 802.15.4 au service des WSN . . . 56
3.3.1 Architecture des WPAN IEEE 802.15.4 . . . 56
3.3.2 Présentation de l'IEEE 802.15.4 . . . 57
3.3.2.1 La couche Physique . . . 57
3.3.2.2 La couche MAC . . . 57
3.3.2.3 La sécurité . . . 59
3.3.2.4 Les couches hautes du modèle OSI . . . 61
3.3.3 Le ZigBee . . . 62

3.3.4 6LoWPAN . . . 64
3.3.5 Les protocoles utilisés dans 6LoWPAN . . . 67
3.3.6 Etats de l'art des systèmes d'exploitation existants . . . 71
3.3.7 Contiki OS . . . 71
3.3.8 Les nœuds de capteurs utilisés . . . 72
3.4 Conclusion . . . 73

4 Exploitation des métadonnées collectées par écoute passive 75
4.1 Introduction . . . 75
4.2 Analyse du réseau ZigBee . . . 76
4.2.1 Le réseau et l'intercepteur déployés . . . 76
4.2.2 Exploitation des métadonnées du réseau ZigBee sécurisé . . . 77
4.2.2.1 Description des expériences . . . 77
4.2.2.2 Analyses des fuites d'information et exploitations . . . 78
4.3 Analyse du réseau 6LoWPAN . . . 81
4.3.1 Le réseau et les outils d'analyse déployés . . . 81
4.3.2 Exploitation des métadonnées du réseau 6LoWPAN sécurisé . . . 82
4.3.2.1 Description des expériences . . . 82
4.3.2.2 Analyses des fuites d'information et exploitations . . . 82
4.4 Analyse des métadonnées du standard IEEE 802.15.4 sécurisé . . . 84
4.5 Conclusion . . . 86

5 Solutions de l'état de l'art pour dissimuler ses identifiants 89
5.1 Introduction . . . 89
5.2 Les techniques d'anonymat . . . 90
5.3 L'utilisation de pseudonymes . . . 93
5.3.1 Les listes de pseudonymes . . . 93
5.3.2 Les pseudonymes dynamiques . . . 97
5.4 Conclusion . . . 100

6 Solutions visant à préserver la confidentialité des adresses 103
6.1 Analyse de MT6D . . . 103
6.1.1 Présentation de MT6D . . . 103
6.1.2 Analyse théorique . . . 104
6.2 Cahier des charges pour une solution idéale . . . 106
6.3 Ephemeral . . . 106
6.3.1 Présentation . . . 106
6.3.1.1 Génération des pseudonymes . . . 107
6.3.1.2 Vérification des pseudonymes . . . 108
6.3.1.3 Analyse de la protection de la vie privée . . . 109
6.3.2 Adéquation d'Ephemeral avec les spécifications listées dans 6.2 . . . 110
6.4 Conclusion . . . 111

7 Evaluation d'Ephemeral 113
7.1 Simulation . . . 113
7.1.1 Environnement de simulation . . . 113
7.1.2 Scénario type . . . 114
7.1.3 Analyse comportementale . . . 116
7.2 Evaluation . . . 117
7.2.1 Réseau de référence . . . 117
7.2.2 MT6D . . . 118
7.2.3 Ephemeral . . . 120
7.2.4 Synthèse . . . 121
7.3 Déploiement . . . 122
7.3.1 Environnement . . . 122

7.3.2 Outils de surveillance du réseau . . . 123
7.3.3 Estimation des performances d'Ephemeral . . . 124
7.3.4 Comportement des nœuds . . . 129
7.4 Conclusion . . . 130

8 Conclusion et perspectives **133**
8.1 Contributions de la thèse . . . 133
8.2 Perspectives . . . 134

A Le WiFi **145**

B Le Bluetooth **149**

C Implémentation de MT6D dans Contiki **153**
C.1 Implémentation dans Contiki 3.0 . . . 153
C.2 Choix d'implémentation pour le déploiement de MT6D dans un réseau multi sauts . . . 154

D Implémentation d'Ephemeral dans Contiki **157**
D.1 Exemple d'utilisation d'Ephemeral . . . 157
D.2 Implémentation dans Contiki 3.0 . . . 158
D.3 Cas critiques . . . 160
D.3.1 Impact sur la QoS . . . 160
D.3.2 Mise à jour de R et problèmes de compteurs . . . 161
D.3.3 Cohabitation avec les mécanismes déjà déployés . . . 161

Table des figures

2.1 Topologie multi sauts. 30
2.2 Modèle OSI adapté aux objets. 31
2.3 Sécurité par couche. 44
2.4 Fonctionnement d'un réseau de mélange. 46
2.5 Solution de vie privée à k-1 sauts. 48
2.6 Schéma d'utilisation des labels. 49

3.1 Comparaison des consommations d'énergie des standards existants. 54
3.2 Format général des trames MAC IEEE 802.15.4. 58
3.3 Mécanismes de sécurité du standard IEEE 802.15.4. 60
3.4 Champs Auxilliary Security Header (ASH). 60
3.5 Format des IV pour le chiffrement. 61
3.6 Modèle OSI du standard ZigBee. 62
3.7 Protocole du standard ZigBee. 62
3.8 Chiffrement et authentification grâce aux mécanismes définit dans ZigBee. 63
3.9 Modèle OSI du standard 6LoWPAN. 65
3.10 Création d'une adresse avec le procédé DHCPv6. 68
3.11 Création d'une adresse avec le procédé SLAAC. 68
3.12 Exemple de DODAG. 70
3.13 Pile µIP de Contiki. 72
3.14 Matériels utilisés. 73

4.1 Plateforme ZigBee déployée. 76
4.2 Protocole de Join ZigBee. 79
4.3 Format simplifié d'une trame ZigBee Network. 80
4.4 Protocole d'association 6LoWPAN. 83
4.5 En-tête MAC disponible en clair par écoute passive. 86

5.1 Anonymat par TOR. 91
5.2 IPsec en mode tunnel. 92
5.3 Attribution d'une liste d'adresse. 94
5.4 Format d'une adresse IPv6 temporaire créée avec la RFC 4941. 96

6.1 Problème de désynchronisation de MT6D. 105
6.2 Structure d'un paquet (simplifiée). 107
6.3 Structure d'un paquet Ephemeral. 108

7.1 Réseau simulé. 114
7.2 Topologie utilisée. 115
7.3 Communications dans le WSN. 115
7.4 Tables du nœud routeur A. 118
7.5 Tables du nœud routeur A pour MT6D. 119
7.6 ICMPv6 vs UDP. 122

7.7 Smart office. 123
7.8 Comparaison de l'empreinte mémoire d'Ephemeral. 127
7.9 Mesure de courant pour l'émission d'une trame RPL. 128

B.1 Adresse BD_ADDR. 149
B.2 Format des trames BR et EDR. 150

D.1 Exemple d'utilisation d'Ephemeral. 157

Liste des tableaux

2.1 Attaques de sécurité par couche OSI. 35

3.1 Comparaison des performances des standards. 54
3.2 Comparaison des 5 standards utilisés dans l'IoT. 55
3.3 Niveaux de sécurité. 59

5.1 Comparaison des différentes méthodes. 102

7.1 Nombre de trames en 20 min. 121
7.2 Temps de calcul. 126
7.3 Comparaison de l'énergie consommée. 128

D.1 Probabilité de collision. 159

Glossaire

6BR 6LoWPAN Border Router.
6LoWPAN IPv6 Low power Wireless Personal Area Network.

AES Advanced Encryption Standard.
AES-CCM Advanced Encryption Standard-Counter with CBC-MAC.
AES-CCM* Advanced Encryption Standard-Counter with CBC-MAC.
AH Authentication Header.
AODV Ad-hoc On-demand Distance Vector.

CAS Cryptographic Anonymity Scheme.
CGA Cryptographically Generated Addresses.
CH Cluster Head.
CM Cluster Member.
CoAP Constrained Application Protocol.

DAD Duplicate Address Detection.
DAO Destination Advertisement Object.
DHCPv6 Dynamic Host Configuration Protocol version 6.
DHT Distributed Hash Table.
DIO DODAG Information Object.
DIS DODAG Information Solicitation.
DODAG Destination Oriented Directed Acyclic Graphs.
DoS Denial of Service.
DTLS Datagram Transport Layer Security.

ECC Elliptic Curve Cryptography.
ESP Encapsulating Security Payloads.
ETX Expected Number of Transmissions.
EUI Extended Unique Identifier.

FFD Full Function Device.

HTTP Hypertext Transfer Protocol.

ICMPv6 Internet Control Message Protocol version 6.
IDS Intrusion Detection System.
IEEE Institute of Electrical and Electronics Engineers.
IETF Internet Engineering Task Force.

IID Interface IDentifier.
IoT Internet of Things.
IP Internet Protocol.
IPsec Internet Protocol security.
IPv6 Internet Protocol version 6.
IV Initialization Vector.

LoWPAN Low power Wireless Personal Area Network.
LR-WPAN Low Rate - Wireless Personnal Area Network.

MAC Message Authentication Code.
MD5 Message Digest 5.
MIC Message Integrity Check.
MITM Man In The Middle.
MT6D Moving Target IPv6 Defense.

NDP Node Discovery Protocol.
NTP Network Time Protocol.

OF Objective Function.
OS Operating System.
OSI Open Systems Interconnection.
OUI Organizationally Unique Identifier.

RFC Request For Comments.
RPL Routing Protocol for Low-Power and Lossy Networks.
RSA Rivest Shamir Adleman.
RSSI Received Signal Strength Indication.
RT-ToF Round Trip - Time of Flight.

SA Security Association.
SAS Simple Anonymity Scheme.
SEND SEcure Neighbor Discovery.
SHA-1 Secure Hash Algorithm 1.
SLAAC StateLess Address AutoConfiguration.
SSAS Simple Secure Addressing Scheme.
SSL Secure Sockets Layer.

TCP Transmission Control Protocol.
TKIP Temporal Key Integrity Protocol.
TLS Transport Layer Security.
TOR The Onion Routing.

UDP User Datagram Protocol.

VPN Virtual Private Network.

WPAN Wireless Personal Area Network.
WSN Wireless Sensor Network.

Publications

Ce travail de thèse a donné lieu à 3 papiers :

- C. Hennebert et J. Dos Santos, *Security protocols and privacy issues into 6lowpan stack : A synthesis*, IEEE Internet of Things Journal, 2014.
- J. Dos Santos, C. Hennebert et C. Lauradoux, *Preserving privacy in secured ZigBee wireless sensor networks*, IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015.
- J. Dos Santos, C. Hennebert, C. Lauradoux et JC Fonbonne, *Ephemeral : Lightweight pseudonyms for 6LoWPAN MAC addresses*, IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016.

Et deux présentations à Connect Security World (CSW) :

- J. Dos Santos, C. Hennebert et C. Lauradoux, 'Privacy issues in 6LoWPAN wireless sensor networks', Septembre 2015.
- J. Dos Santos, C. Hennebert et C. Lauradoux, 'Ephemeral privacy analysis', Septembre 2016.

Partie 1

Introduction générale

1.1 L'IoT

Le terme Internet of Things (IoT) est connu depuis plus d'une décennie. Néanmoins, sa démocratisation auprès du grand public ainsi que l'intérêt que lui porte les industriels et les chercheurs ne sont que très récents. L'IoT peut être défini comme le concept qui permet de connecter dans un même réseau global des appareils ou objets divers (capteurs, ordinateurs, appareils électroniques...), identifiables de manière unique, grâce à des communications filaires et sans fil.

L'IoT crée un nouvel environnement où chaque objet peut communiquer directement avec les autres objets composants le réseau. Il permet également de lier le monde physique et le monde digital. Il rend ainsi accessible à n'importe quel utilisateur les données du monde physique. L'IoT étend donc le monde d'Internet et du Web au monde physique par l'intermédiaire de ces nouveaux objets intelligents. Ce nouveau concept permet la mise en place de réseaux coopératifs mais également crée de nouvelles perspectives d'applications.

Les exemples sont nombreux et les domaines d'applications tout autant. Dorge et al. dans l'article [1] décrivent l'utilisation de l'IoT pour la mise en place d'un système de domotique intelligent relié à un serveur Web permettant le contrôle du chauffage, de l'éclairage ou encore de l'ouverture des portes. Dohare et al. décrivent dans l'article [2] un système IoT permettant la surveillance d'une mine de charbon afin de sécuriser le chantier. Les données sur la mine sont collectées et partagées en temps réel entre les mineurs et les superviseurs restés à l'extérieur de la mine afin de prévenir rapidement d'un danger.

L'un des points communs de nombreuses de ces applications est la nécessité de collecter des informations sur l'environnement physique. La collecte de mesures physiques est effectuée par des capteurs, terme consacré par abus de langage aux objets de l'IoT, qui interagissent avec le monde physique. La mise en place de réseau de capteurs Wireless Sensor Network (WSN) permet le déploiement de capteurs dans des environnements vastes et contribue ainsi au développement de l'IoT et à la diversité de ses domaines.

1.2 Les réseaux de capteurs

Avec l'IoT, le besoin de connaitre et de surveiller le monde extérieur est de plus en plus important. Les données collectées doivent ensuite pouvoir être échangées rapidement (quasiment en temps réel) et envoyées vers un réseau informatique.

La mise en place de WSN est donc devenue un domaine de recherche très abordé. Les réseaux de capteurs font parti des réseaux Low power Wireless Personal Area Network (LoWPAN). Ces réseaux sont composés d'un grand nombre d'éléments, appelé nœud, communicants sur des courtes distances grâce à un canal sans fil. Ils collaborent afin de collecter les données physiques et de remonter ces données vers un ou plusieurs points de collecte mais également d'interagir avec leur environnement comme les actionneurs.

Ces nœuds de capteurs sont des systèmes embarqués intelligents comportant plusieurs éléments :

- L'unité de mesure. Elle permet de collecter la grandeur physique de l'environnement comme la température ou l'humidité. Un nœud peut embarquer plusieurs capteurs.
- L'unité de traitement. Elle est composée d'un processeur et d'une mémoire.
- Une alimentation. Généralement, les nœuds fonctionnent sur batteries ou avec des alimentations limitées (panneau solaire, système de récupération d'énergie ...).
- Une radio pour communiquer. Ce module est sans fil.

L'architecture de ces nœuds a été pensée dans un objectif de limiter les coûts de fabrication et de permettre un déploiement large échelle et varié. Ils possèdent une mémoire et une puissance de calcul limitées et fonctionnent avec une source énergétique limitée.

Ainsi, ces nœuds sont petits et bon marché ce qui permet de les intégrer facilement et massivement dans les différents environnements. Les applications basées sur les WSN proposent le déploiement dense et aléatoire d'un grand nombre de ces nœuds.

Suivant l'application recherchée, les nœuds peuvent être déployés dans des environnements hostiles mais également accessibles à n'importe qui telle qu'une ville. Le déploiement peut se faire de manière aléatoire (lancés d'un avion) ou suivant un plan défini.

Dû au grand nombre de nœuds déployés ainsi qu'à la difficulté d'accéder à certains d'entre eux, le changement de piles est impraticable ou très coûteux. Les nœuds doivent donc posséder une durée de vie importante. L'estimation espérée est d'environ 10 ans. Les nœuds ainsi que les protocoles déployés dans le réseau doivent donc être basse consommation.

La contrainte énergétique est donc un grand challenge pour le déploiement des WSN. De nombreuses recherches ont été menées au niveau protocolaire afin d'économiser l'énergie en réduisant au minimum les échanges nécessaires au fonctionnement de ces protocoles.

Des mécanismes permettant de limiter les collisions ont tout d'abord été implémentés comme celui décrit par Rajendran, Obraczka et al. dans [3]. Ils proposent *TRaffic-Adaptative Medium Access* (TRAMA), un mécanisme d'anticollision pensé pour les WSN et permettant de limiter l'énergie consommée. Dans leur solution, les communications dans le réseau doivent être organisées via l'utilisation de slots temporels. Les informations sur le trafic de chaque nœud sont ensuite utilisées pour déterminer l'ordre des communications permettant de ne pas allouer de slot à des nœuds ne souhaitant pas communiquer. Le protocole CSMA-CA qui permet d'éviter les collisions en testant l'occupation de la radio avant d'émettre est également conseillé par de nombreux standards.

Un effort a également été réalisé au niveau du contrôle du module radio. En effet, lors de communications sans fil, ce sont les étapes d'émission et de réception qui sont les plus énergivores. Des mécanismes de veille sont donc utilisés. Ils permettent de maximiser le temps de veille de la radio et ainsi de limiter le temps où le nœud écoute le canal. Le but recherché est d'atteindre 99% de temps de veille.

Les WSN possèdent des similitudes avec les réseaux ad hoc mais contrairement à eux, les nœuds sont autonomes, auto organisés ce qui permet de réduire l'intervention humaine nécessaire au bon fonctionnement du réseau.

Ce comportement implique l'utilisation de protocoles permettant le déploiement et la configuration automatiques du réseau. Les communications sans fil sont incertaines. Des pertes de paquets peuvent apparaitre ou des paquets peuvent être endommagés. De plus, les WSN n'autorisent pas le renvoi d'une trame perdue. Ce comportement doit donc également être pris en compte par les différents protocoles présents dans le réseau.

Le réseau est donc conçu pour s'auto configurer et s'auto organiser. Les nouveaux arrivants doivent réussir à rejoindre le réseau et communiquer avec les nœuds déjà déployés et ce sans intervention humaine. Pour cela, des mécanismes de découverte de voisins doivent être utilisés. Ils permettent ainsi de s'annoncer aux nœuds déjà présents mais également d'échanger les différentes informations nécessaires aux communications dans le réseau.

La topologie du réseau n'est pas fixe et peut être modifiée par l'ajout ou la suppression de nœuds. L'ajout de nouveaux nœuds n'est pas surveillé par un utilisateur. Contrairement aux réseaux Internet Protocol (IP) classiques, un grand nombre de ces nœuds peuvent être mobiles ou ajoutés après déploiement. Les protocoles de maintien et de routage déployés doivent donc s'adapter à ces nouvelles contraintes. Ils sont dépendants de l'application et de l'architecture du WSN. Le protocole de routage ne peut utiliser une table pré enregistrée. Il faut qu'il puisse mettre à jour dynamiquement sa connaissance de la topologie tout en prenant en compte les pertes possibles de paquets. Pour cela, le protocole doit être capable d'interroger les nœuds du réseau afin de connaitre les différents participants et d'estimer le meilleur chemin pour remonter son information collectée tout en limitant la consommation d'énergie. De nombreux protocoles ont déjà été publiés. Al-Karaki et al. détaillent dans [4] les différents protocoles de routage existants (en 2004) dans les WSN. Ils classent les solutions de l'état de l'art en trois catégories suivant la topologie du réseau et comparent les avantages de chaque classe.

1.3 La sécurité des WSN : un challenge actuel

Les différentes applications disponibles avec les WSN et la criticité de certaines d'entre elles rendent la sécurité d'autant plus nécessaire. Les caractéristiques de ces réseaux offrent de nouvelles possibilités d'attaques. Le média sans fil ainsi que le déploiement en environnement ouvert facilitent les attaques et l'accès au réseau. La nature contrainte des nœuds notamment en énergie donne un avantage aux attaquants. La topologie et le déploiement étant aléatoires, il est compliqué d'utiliser des techniques de surveillances basées sur ces critères et la connaissance a priori du réseau. L'attaquant peut alors exploiter le manque de supervision du réseau et de son déploiement pour mener ses attaques. Assurer la sécurité dans ces conditions est plus difficile que pour les réseaux classiques et des solutions sur mesure doivent être introduites.

Les recherches sur les WSN ayant été focalisées sur les protocoles et les problèmes techniques précédents, la mise en place de la sécurité a été quelque peu délaissée. Néanmoins, depuis quelques années la tendance tend à s'inverser et des recherches sont menées afin d'assurer la sécurité dans les WSN tout en tenant compte de la consommation énergétique supplémentaire.

Ainsi la cryptographie et les mécanismes de sécurité classiques ont dus s'adapter à ces nouvelles contraintes. La confidentialité mais également l'authentification et l'intégrité des données ont dues être repensées pour fonctionner avec des communications incertaines et facilement accessibles pour un attaquant externe. Des solutions de chiffrement bas coût ont été introduites pour être embarquées dans ces capteurs. La distribution et le renouvellement des clés sont également compliqués dans ces environnements. Malgré ces nouveaux protocoles sécurisés de nouvelles attaques sont menées qui nécessitent des contre mesures adaptées et dédiées. Nous allons étudier ces nouvelles menaces mais également les contre mesures de l'état de l'art dans la partie 2.

L'une des menaces à laquelle doit faire face les réseaux de capteurs concerne la protection de la vie privée. Ces réseaux appartiennent à un propriétaire tel qu'une entreprise comme pour l'exemple de la mine de charbon ou encore à un particulier dans le cadre de la domotique. Ils peuvent également appartenir à une collectivité comme une ville et les données peuvent être mises à disposition des citoyens. Des utilisateurs vont accéder aux données et l'environnement de ces utilisateurs va être surveillé. Il faut donc assurer la vie privée des différents acteurs.

Les données et métadonnées contenues dans les trames échangées et nécessaires au bon acheminement des données sont accessibles facilement pour un attaquant externe. Ces métadonnées collectées permettent de connaitre les données physiques de l'environnement grâce aux capteurs mais également les informations des utilisateurs qui vont consulter ou fournir ces données. Elles permettent également de déduire des informations sur le propriétaire du réseau. Le déploiement de ces WSN facilite la collecte par un attaquant d'informations de vie privée. Apthorpe, Reisman et al. dans [5] analysent les informations de vie privée disponibles pour un attaquant externe passif sur quatre objets domotiques du commerce. Leurs analyses permettent de montrer que même lorsque le trafic est chiffré, la connaissance des métadonnées permet de déduire des informations sur les utilisateurs des différents produits comme les heures où ils se couchent ou encore leur présence/absence et donc sur leurs habitudes.

Il est donc primordial d'assurer la confidentialité des données échangées mais également des métadonnées permettant de transporter l'information utile jusqu'au point de collecte. Des recherches ont donc été menées afin de fournir des solutions de protection. Dans cette thèse, nous ne nous sommes pas intéressés à cette partie de la protection de la vie privée.

Nous nous sommes focalisés sur les informations concernant le réseau et ses nœuds également accessibles et déductibles grâce à la nature ouverte et sans fil du média. Un attaquant a alors accès grâce aux métadonnées aux identités des nœuds mais également aux informations sur le fonctionnement du réseau. Il peut ensuite utiliser les informations collectées pour mener des attaques plus puissantes et contrer les mécanismes de sécurité déployés. Il est donc essentiel de mettre en place une solution évitant au maximum la collecte et l'utilisation de ces métadonnées concernant le réseau. Ce sujet de recherche a été relativement peu traité dans la littérature notamment dans le cadre de réseau de capteurs contraints.

1.4 Organisation du mémoire et contributions

La mise en place de réseaux IoT et plus particulièrement de WSN nécessite de rendre chaque nœud identifiable de manière unique. Cet avantage pour le déploiement de nouvelles applications offre des failles exploitables par un attaquant.

Les études menées se sont focalisées principalement sur la mise en place de solutions de sécurité efficaces et peu coûteuses afin d'éviter ou de prévenir des attaques pouvant nuire au réseau ou à ses performances et fonctionnant avec la nature atypique de ces réseaux. Néanmoins, nous avons constaté que peu de recherches abordent la protection de la vie privée et plus particulièrement la protection des métadonnées échangées lors des communications sans fil amenant à des fuites sur les identifiants et sur les caractéristiques du WSN.

Nous avons étudiés et comparés les différents protocoles utilisés pour le déploiement de WSN mais également les failles de sécurité et de vie privée apportées par chacun d'eux. Nous avons identifiés qu'actuellement l'IEEE 802.15.4 est l'un des standards dont la sécurité est la moins bien assurée et qu'il offre ainsi des fuites d'identifiants notamment les adresses MAC utiles pour le routage, exploitables facilement et efficacement par un attaquant.

Nous avons proposé d'étudier les fuites existantes lors d'écoutes passives des communications dans un réseau IEEE 802.15.4. De notre point de vue, les identifiants MAC uniques accessibles même lorsque la sécurité est activée, représente l'une des fuites d'information les plus utiles pour un attaquant et des solutions doivent être mises en place pour éviter leurs exploitations.

Cette thèse considère les problèmes de protection de la vie privée dans les réseaux de capteurs et plus particulièrement la collecte et l'utilisation des identifiants par un attaquant. Elle fourni une solution adaptée aux réseaux de capteurs contraints basés sur le standard IEEE 802.15.4. Nous proposons Ephemeral solution permettant à des nœuds contraints de générer des pseudonymes dynamiques à l'aide d'une fonction cryptographique et à utiliser à la place des adresses uniques MAC essentielles pour les communications sans fil. Cette solution assure la protection de la vie privée tout en permettant l'ajout mais également la mobilité des nœuds. Aucune autorité n'est nécessaire pour la génération et la vérification des pseudonymes. Elle est compatible avec les différents protocoles déjà présents.

Cette thèse est composée de sept autres parties proposant une solution à la fuite des identifiants lors des communications sans fil IEEE 802.15.4.

La **partie 2** détaille de façon plus précise les vulnérabilités existantes dans les WSN. Elle donne les notions importantes utiles pour la compréhension de la thèse et notamment les outils cryptographiques utilisés dans les WSN. Cette partie présente certaines attaques de sécurité détaillées vis-à-vis de la couche du modèle OSI où elles opèrent ainsi que les contre mesures existantes dans l'état de l'art. Un travail similaire est effectué afin d'identifier les attaques exploitant les métadonnées et les contre mesures existantes. Cette analyse nous permet d'identifier un manque de solutions efficaces pour la protection des adresses dans les WSN.

La **partie 3** présente le standard IEEE 802.15.4 ainsi que les descriptions des modèles ZigBee et 6LoWPAN, modèles basés sur les couches basses IEEE 802.15.4. Le standard IEEE 802.15.4 est tout d'abord comparé aux autres standards utilisés pour l'IoT. Nous détaillons ensuite la couche Physique ainsi que la couche MAC de ce standard puis nous abordons la sécurité proposée. Cette sécurité sera exploitée dans notre solution de protection de la vie privée. Les mécanismes de sécurité proposés dans le standard ZigBee ainsi que dans le standard 6LoWPAN sont également analysés. Les différents protocoles utiles aux déploiements et au maintien des WSN et exploités dans la thèse sont expliqués. Enfin, les dernières parties permettent de présenter les technologies utilisées dans la thèse.

La **partie 4** est dédiée à inférer par écoute passive un maximum d'information sur des réseaux ZigBee et 6LoWPAN sécurisés. Cette partie détaille la mise en place des deux réseaux de test. Les deux outils de collecte utilisés pour mener les attaques passives sont détaillés et le coût de déploiement de chacun est estimé. Les outils de traitement et d'analyse utilisés pour inférer un maximum d'information de vie privée sont également présentés. L'exploitation des métadonnées des deux réseaux est réalisée ce qui permet de déduire les fuites communes liées à l'utilisation pour les couches basses du standard IEEE 802.15.4. Cette partie permet d'identifier et de confirmer que les adresses MAC IEEE 802.15.4 représentent une fuite d'information de vie privée utile pour un attaquant.

La **partie 5** donne l'état de l'art des solutions visant à protéger les identités. Deux catégories de solutions sont abordées. La première permet de fournir l'anonymat. Dans la deuxième catégorie, le choix est fait de permettre l'utilisation de pseudonymes à la place des identifiants présents dans les métadonnées. C'est à cette catégorie qu'appartient la solution de l'état de l'art que nous avons identifiée comme la plus pertinente et la plus adaptée aux besoins des réseaux de capteurs IEEE 802.15.4.

La **partie 6** présente le concept d'Ephemeral, notre solution de protection de la vie privée. Nous avons tout d'abord voulu analyser théoriquement le fonctionnement de la solution de l'état de l'art identifiée précédemment. Cette partie détaille tout d'abord son fonctionnement. Puis les avantages et inconvénients sont identifiés. Cette analyse théorique nous permet de définir un cahier des charges idéal pour une solution de protection des identifiants. Cette partie présente alors Ephemeral et son fonctionnement. Elle détaille la génération mais également la vérification des pseudonymes. Certains problèmes de l'utilisation de fonctions cryptographiques sont étudiés et la robustesse d'Ephemeral vis-à-vis d'eux est validée. Enfin, Ephemeral est analysé théoriquement et son adéquation avec le cahier des charges est détaillée.

La **partie 7** est dédiée à l'évaluation d'Ephemeral. Une première comparaison est réalisée en simulation entre la solution de l'état de l'art, Ephemeral et un réseau de référence sans protection des métadonnées. Cette partie permet de valider l'étude théorique de la partie précédente des deux solutions mais également de positionner Ephemeral vis-à-vis de cette solution et d'identifier ses lacunes ainsi que celles du déploiement et des solutions technologiques choisies. Un déploiement réel de notre solution est ensuite réalisé afin de permettre l'évaluation de paramètres indisponibles en simulation et important dans les réseaux contraints comme la consommation d'énergie ou l'empreinte mémoire. L'impact du déploiement de notre solution de protection des identifiants MAC est donc fourni et quantifié.

La **partie 8** conclut la thèse en résumant les principales contributions. Des pistes d'améliorations et des perspectives d'évolutions sont données.

Partie 2

Les vulnérabilités dans les WSN

Dans cette partie, les problèmes liés à la sécurité et à la protection de la vie privée dans les réseaux de capteurs sont présentés. Les notions importantes utilisées tout au long de cette thèse sont tout d'abord définies. Les attaques contre la sécurité, les différents modèles d'attaquants ainsi que les contre mesures existantes de l'état de l'art sont abordées. Nous montrons que la protection de la vie privée est difficile à assurer dans des réseaux sans fil contraints.

2.1 Introduction

Internet et ses applications ont pris une place de plus en plus importante dans notre quotidien si bien qu'il devient difficile voire impossible de se passer de celui-ci que ce soit dans notre vie professionnelle ou personnelle.

Cette avancée a été rendue possible grâce à la miniaturisation des systèmes embarqués mais également au développement de standards de communication adaptés. L'avènement de l'Internet des objets (IoT), représente une aubaine pour les développeurs d'applications. L'utilisation d'Internet s'est beaucoup démocratisée jusqu'à s'étendre à des objets auxquels personne n'aurait pensé.

Plus particulièrement, le déploiement de capteurs et des réseaux associés WSN a changé notre façon d'interagir avec notre environnement. Ils ont permis la collecte de données dans des environnements où les réseaux filaires ne pouvaient être déployés. La collecte et l'analyse en temps réel des données de notre environnement permettent la mise en place de nouvelles applications :

- Domotique. Les capteurs sont de plus en plus déployés au sein des habitations personnelles ou des locaux collectifs. Une surveillance et une gestion de la consommation des ressources (électricité, chauffage...) dans un but économique et écologique sont devenues primordiales. Des applications peuvent également utiliser les données collectées par les différents capteurs afin d'améliorer la qualité de vie et le confort des habitations. Un habitant peut ainsi contrôler à distance la consigne de température de son logement afin de s'assurer que celle-ci soit parfaite pour son retour à son domicile. Il existe de nombreuses solutions dites "prêtes à l'emploi" vendues par les grands Fournisseurs d'Accès à Internet (FAI) comme Orange et son système Homelive dont l'entrée de gamme se situe à 79 € moyennant ensuite un abonnement mensuel de 9,99€. Le pack inclut un détecteur d'ouverture de porte, un détecteur de mouvement et une prise intelligente. Leur système permet le déploiement d'un détecteur d'intrusion en plus de la gestion énergétique. Néanmoins, avec la démocratisation des matériels bas coût et la culture du *Do It Yourself* (DIY), il est maintenant plus aisé de réaliser soi-même son propre système domotique. Butt, Zulqarnain et al. dans [6] proposent un système contrôlable via une application mobile utilisant un Raspberry Pi pour le contrôle et plusieurs capteurs. Leur système permet des applications similaires à celles de Homelife.

- Localisation. Un utilisateur peut via une application sur smartphone demander à ses lunettes de vue connectées d'émettre un signal afin qu'il puisse les retrouver. Ce système de localisation peut être déployé sur un grand nombre d'objet comme des clés ou encore des télécommandes. De nombreuses technologies basées sur les Impulsions Radio Ultra-Wide-Band (IR-UWB) ont été déployées comme celle utilisée par la société Bespoon. Néanmoins, la précision des différentes technologies n'est pas identique. Selon l'étude menée par Wang, Raja et al. dans [7] sur trois technologies présentes dans le commerce, cette précision peut aller de 20 cm à 2 cm pour la technologie Bespoon. Le besoin de l'application influe donc sur le choix de la technologie.
- Applications médicales. Internet permet également de rassembler au sein d'une base de données des informations collectées par des capteurs. Le but est d'assurer un suivi régulier d'un patient souffrant de problèmes cardiaques sans besoin d'accéder au cabinet de son cardiologue. Les capteurs sont alors déployés sur le patient. On parle de réseaux *Wireless Body Area Network* (WBAN) défini dans le standard IEEE 802.15.6 [8].
- Les villes intelligentes. Grâce aux WSN, il est possible d'améliorer notre quotidien et celui de nos concitoyens. Les divers capteurs embarqués dans les smartphones, tablettes ou véhicules des personnes permettent une collection massive de données au sein d'une ville. Ces nouvelles données devraient selon [9] changer le développement de nos villes en permettant de créer de nouvelles applications/services qui simplifieront la vie de ses habitants. Les avantages de la géo localisation ont particulièrement été considérés dans [9]. L'utilisation du GPS, Bluetooth ou d'autres technologies permet de fournir des services adaptés. Les traces peuvent être utilisées dans le cadre d'un service de navigation intelligent. Les GPS des véhicules vont alors fournir des informations de densité et de condition de trafic prises en compte pour recommander le meilleur itinéraire et désengorger les routes. Les informations de localisation des personnes peuvent également aider à la gestion des épidémies. En effet, l'historique des déplacements d'une personne atteinte de la grippe pourra être analysé afin de comprendre et d'estimer la propagation de l'épidémie et donc prévoir en conséquence les services médicaux.

La diversité des technologies, des standards déployés et des applications fait de l'IoT un domaine de recherche encore à explorer. Si ces nouvelles technologies sont pleines de promesses, il faut néanmoins considérer leurs "effets secondaires". En effet, elles impliquent une collecte massive de données qui peut être préjudiciable à la vie privée de l'utilisateur. Les objets connectés interagissent avec les différents services via des réseaux hétérogènes sensibles à de nombreuses vulnérabilités.

Dans le rapport [10], Kowatsch et al. analysent l'acceptation d'un service IoT par un panel d'utilisateurs. Ils en déduisent que pour que celui-ci soit pleinement accepté, la législation entourant la collecte et l'exploitation de ces données mais également la sécurité des données et la connaissance par un utilisateur du type d'informations qu'il partage vont jouer un rôle important. Certaines applications selon [10] informent quelles données concernant l'utilisateur sont collectées et dans quel but. Néanmoins, dans certains cas, les applications collectent des informations sur les utilisateurs sans les en avertir ni donner des explications sur l'utilité. Cette collecte peut avoir lieu même lorsque l'application tourne en "tâche de fond" car l'utilisateur ne l'a pas fermée. La législation doit alors assurer aux utilisateurs que l'application les informe de toute collecte ainsi que le but de celle-ci.

L'utilisateur ne veut pas que son identité ou toutes informations personnelles soient divulguées à son insu et veut pouvoir faire confiance à une autorité pour gérer ses informations.

Le projet RERUM [11] (REliable, Resilient and secUre iot for sMart city applications) part du principe que pour assurer l'adoption d'un standard et donc pour le déploiement d'un réseau IoT, certains critères doivent être respectés :

- La sécurité.
- La protection de la vie privée.

Ceux-ci doivent être assurés en tout point de la chaine d'information, de la collecte à l'utilisation.

Or, les WSN sont fortement contraints en énergie, en portée mais également en mémoire. Les communications sont réalisées avec un canal sans fil ce qui les exposent à de nombreuses vulnérabilités. Du fait de ces contraintes, il est difficile de garantir la sécurité et la protection de la vie privée au sein même du WSN. C'est pour ces raisons que j'ai travaillé durant ma thèse sur cette thématique.

Afin de mettre en place le bon mécanisme de défense, il est nécessaire de tenir compte du type d'attaquant, de ses capacités ainsi que de l'attaque qu'il va vouloir mener. Il est donc essentiel de modéliser l'attaquant afin de fournir la contre mesure la mieux adaptée.

Le déploiement et l'acceptation de l'IoT passent également par la mise en place de solutions répondant aux critères précédents et compatibles avec le matériel et les standards existants.

Enfin, ces critères doivent bien sûr respecter le bon fonctionnement du réseau et ne pas trop lui nuire. En effet, l'introduction de mécanismes assurant ces critères entraine inévitablement des pertes de performances. Néanmoins, comme l'expliquent Pohls et al. dans [11], il est nécessaire de trouver le bon compromis entre une collecte des données efficace, pseudo temps réel effectuée par des capteurs de confiance et une transmission sécurisée et privée de celles-ci.

Je vais d'abord présenter succinctement les réseaux de capteurs puis l'apport de la cryptographie à leur sécurité. J'exposerai par la suite les attaques qui ne sont pas gérées par la cryptographie puis je présenterai les problèmes de vie privée.

2.2 Les réseaux de capteurs

Afin de réaliser les différentes applications données précédemment, il existe deux moyens d'interagir avec les réseaux de capteurs.
Dans le premier cas, les nœuds du réseau sont interrogés afin d'obtenir les données collectées. On parle de communications "point-à-multi-point". Les requêtes entrent et sortent du réseau par un unique point d'entrée appelé *gateway* ou passerelle.
La *gateway* est un élément clé des réseaux de capteurs. Elle implémente les couches de communications de différents standards afin de relier les différents réseaux. Elle va permettre de rendre intelligible par un réseau une donnée formatée par un autre réseau. De nombreuses recherches ont été menées pour optimiser sa consommation d'énergie mais également permettre l'interopérabilité des standards de communications.
L'autre méthode d'interaction avec le réseau consiste à autoriser les nœuds à transmettre périodiquement leurs données vers un nœud plus performant appelé nœud puits. Dans ce cas d'utilisation, on parle de communications "multi-point-à-point". Le déclenchement de l'envoi peut être dû à un *timer*, on parle alors de *Time-driven* ou à un phénomène observé c'est alors une application *Event-driven*. Ces données peuvent ensuite être envoyées vers la *gateway* pour une utilisation depuis le Web. Dans ce cas, la *gateway* et le nœud puits communiquent directement.

Des topologies multi sauts comme celle de la Figure 2.1 sont déployées. Elles permettent de pallier la faible portée des communications en autorisant des nœuds dits routeurs (nœud bleu) à remonter les données collectées par un autre nœud (nœud vert) vers le nœud puits (nœud rouge) ou la *gateway*. Le nœud vert peut lui aussi être routeur.

Ce nœud puits numéroté 0 sur la Figure 2.1 joue également un rôle plus important. En effet, les WSN ont besoin d'un nœud plus puissant et moins contraint afin de gérer la mise en place du réseau mais également son maintien. Ce nœud appelé "racine" ou *cluster head* ou encore "station de base" permet également la gestion des protocoles de sécurité. De ce fait, il ne peut fonctionner sur batterie et doit posséder une mémoire plus importante. Il est également possible de réunir le rôle de racine dans la *gateway*. Le réseau possède alors un seul nœud puissant jouant le rôle de *gateway*, de racine et de puits.

Afin d'assurer un bon fonctionnement, les nœuds ont donc besoin de protocoles leur permettant de s'auto configurer et de s'auto déployer suivant différentes topologies multi sauts. Le réseau a besoin de découvrir tous ses participants, organiser les communications, le routage et de permettre aux nœuds d'être adressables depuis l'extérieur. Ces tâches peuvent être compliquées à cause des contraintes dans les WSN. Il faut qu'un nœud puisse se configurer même en présence de pertes de communication ou lorsqu'il se trouve éloigné du nœud puits.

Une première étape consiste à obtenir pour chaque nœud une adresse unique valide dans le réseau pour communiquer. Ce protocole d'adressage est très important pour le WSN. Assurer la configuration rapide ainsi que l'unicité d'une adresse dans un réseau vaste et contraint peut être énergivore.

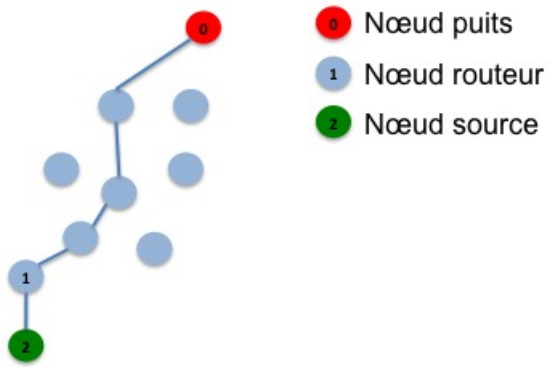


FIGURE 2.1 – Topologie multi sauts.

De manière classique, les nœuds interrogent un serveur afin d'obtenir une adresse valide et unique. Dynamic Host Configuration Protocol version 6 (DHCPv6) [12] utilise cette méthodologie et permet à un nœud d'obtenir une adresse Internet Protocol version 6 (IPv6). Bien que permettant d'obtenir de manière certaine une adresse unique, cette méthode est compliquée à déployer dans les WSN et peut rendre la durée du protocole d'adressage longue voire rendre le protocole d'adressage impossible. D'autres méthodes ont alors été introduites permettant la configuration rapide mais nécessitant la détection de l'unicité afin d'éviter les collisions. Le protocole Duplicate Address Detection (DAD) introduits dans la Request For Comments (RFC) 2462 [13] permet de détecter la duplication d'une adresse. Un nœud va, dès lors où il possède une nouvelle adresse, en avertir ses voisins. Pour cela, il va émettre des trames supplémentaires contenant son adresse. Si l'un des nœuds du réseau possède une adresse identique, il va alors répondre au nœud émetteur, annulant l'adresse et obligeant ce dernier à recommencer le protocole d'adressage. Dans un réseau vaste, la propagation dans tout le réseau prend du temps et consomme beaucoup d'énergie. Les trames peuvent se perdre et il est donc compliqué d'assurer cette unicité rapidement et efficacement.

Les WSN sont déployés dans des environnements hostiles où l'intervention humaine est difficile voire impossible. Ils sont généralement vastes et contiennent de nombreux nœuds. Les données collectées ont alors besoin d'être routées jusqu'à leur destination finale. Contrairement aux réseaux classiques, la structure des WSN peut évoluer rapidement suite à la perte de connexion d'un nœud ou dû à du mouvement dans le réseau. L'algorithme de routage doit donc prendre en compte ces contraintes. Le choix du protocole de routage le plus optimal est devenu en quelques années un sujet de recherche très abordé. L'étude faite par Goyal et al. dans [14] décrit les différents protocoles de la littérature. Dans ceux-ci, un arbre de routage est créé afin de connaitre les routes et d'acheminer convenablement les données. Des relations père/fils sont établies. Dans la Figure 2.1, le nœud 1 joue le rôle de père pour le nœud 2. Réciproquement le nœud 2 est le fils de 1. Lorsque le fils veut communiquer, il doit transférer sa donnée collectée à son père pour un routage vers le nœud puits 0.

Chaque nœud à donc besoin d'avoir une vision de son voisinage pour router les données et choisir le chemin optimal. Il lui faut connaitre l'adresse de son père mais également de son ou ses fils. Deux méthodes de découvertes de chemin existent.
Dans la méthode réactive, le nœud calcule le chemin de routage avant chaque envoi. La RFC 3561 [15] introduit Ad-hoc On-demand Distance Vector (AODV), un protocole réactif. Un nœud qui souhaite communiquer va émettre des paquets broadcastés dans tout le réseau afin de déterminer le chemin optimal à l'instant t. Grâce aux réponses obtenues, il construit alors le chemin de routage.
Dans un schéma proactif comme Optimized Link State Routing Protocol (OLSR) [16], des tables de routage sont maintenues. Les informations nécessaires sont collectées lors des protocoles de découverte de voisins. Néanmoins, contrairement au réseau filaire, la topologie peut évoluer, les nœuds se déplacer et par conséquence, le voisinage du nœud peut changer. Il est alors nécessaire de permettre aux nœuds d'effectuer ce processus périodiquement afin de mettre à jour ces données. Ce protocole appelé *Hello* doit être le plus simple possible pour ne pas consommer trop d'énergie. Il est également utilisé par un nouveau nœud souhaitant rejoindre le réseau. On parle alors de phases de *join* et d'association.

Afin d'assurer les communications aux seins du WSN, les nœuds implémentent le modèle Open Systems Interconnection (OSI) de la Figure 2.2. Ce modèle est adapté aux contraintes des WSN et comporte 5 couches au lieu des 7 habituelles. Il permet de définir ce que chaque couche doit gérer. Pour cela, les différents organismes de standardisations publient des normes pour chacune d'entre elles.

FIGURE 2.2 – Modèle OSI adapté aux objets.

La couche Physique permet de spécifier les caractéristiques matérielles mais également celles du signal (modulation, fréquences...). Elle est publiée dans le même standard que la couche 2.
La couche MAC ou couche liaison permet d'indiquer comment les données vont être émises entre deux nœuds voisins. Elle gère le contrôle d'erreurs dues aux transmissions, l'accès au média ou encore la formation du paquet. Dans le cas d'un routage *mesh-under*, la couche MAC peut intégrer le mécanisme de routage.

Au-dessus, se trouve la couche Réseau. Elle permet de gérer le routage des données dans un réseau multi saut pour un routage *route-over*. La route établie doit optimiser l'énergie consommée par les capteurs et les temps de latences pour le transport des données. Cette couche définit également le processus d'adressage.

La couche Transport permet de gérer le découpage et la réorganisation des paquets, le contrôle de flux. Transmission Control Protocol (TCP) et User Datagram Protocol (UDP) sont deux exemples de mécanismes possibles pour la couche Transport. Enfin, la couche Application assure l'interface avec les applications. Hypertext Transfer Protocol (HTTP) est l'un des protocoles possibles de cette couche.

Lors d'une communication, la donnée utile (la valeur de la température par exemple) est encapsulée dans différents en-têtes ajoutés de manière successive par les différentes couches traversées. Ces en-têtes contiennent des informations appelées métadonnées qui permettent au récepteur de traiter la trame et d'en connaitre le format. Lors de la réception, chaque couche enlève et traite l'en-tête lui correspondant et transmet le paquet à la couche suivante.

2.3 Apport de la cryptographie à la sécurité des WSN

La cryptographie permet d'apporter les propriétés suivantes dans les communications des WSN :

- la confidentialité. Une information ne doit être révélée qu'aux personnes légitimes et concernées par les communications.
- l'authentification. Il est possible de vérifier l'identité des personnes qui émettent des messages.
- le contrôle d'accès. Propriété proche de l'authentification, le contrôle d'accès permet de ne donner l'accès aux ressources qu'aux utilisateurs légitimes.
- l'intégrité. Il est possible de détecter les modifications, délibérées ou non, des communications.
- la non répudiation. Un nœud ne peut a posteriori nier l'envoi d'un message.

Dans l'article [17], Wallgren, Raza et al. analysent les technologies de l'IoT et les problèmes de sécurité notamment en terme d'attaques sur le protocole de routage. Ils démontrent via le déploiement de réseaux simulés l'importance de la sécurité dans les WSN. Ils expliquent que le mécanisme assurant la confidentialité est le premier outil à mettre en place.

2.3.1 La confidentialité

La confidentialité est l'un des premiers objectifs de la cryptographie. Le problème est le suivant :
deux utilisateurs Alice et Bob souhaitent communiquer sans qu'un troisième individu nommé Eve, qui écoute

la conversation, puisse en connaitre le contenu. Pour cela, Alice va utiliser un algorithme de chiffrement afin de modifier le message à envoyer à Bob et le rendre inintelligible pour Eve. Bob doit être capable de retrouver le message initial. Les deux transformations de chiffrement et déchiffrement dépendent de clés inconnues d'Eve. Deux types de chiffrement sont alors possibles.

Dans un premier schéma, la clé est commune à Alice et Bob et a été préalablement échangée. On parle alors de chiffrement symétrique. Cette clé n'est pas connue d'Eve et permet à Bob de déchiffrer le message chiffré par Alice. Ces algorithmes sont rapides. Deux familles de chiffrements symétriques existent.
Les algorithmes à flot fonctionnent bit à bit. Ils génèrent à partir de la clé k une suite de symboles. Cette suite chiffrante possède une taille identique à celle du message à chiffrer. Le message en clair est alors combiné bit à bit à la suite chiffrante pour obtenir le message chiffré.
La deuxième catégorie est le chiffrement par bloc. Comme son nom l'indique, elle divise les données à chiffrer en blocs de taille identique. Le chiffrement des différents blocs s'effectue selon des modes opératoires. L'Advanced Encryption Standard (AES) est l'algorithme de chiffrement standardisé par le NIST et qui est le plus utilisé dans le monde industriel.

Dans le deuxième schéma, la cryptographie repose sur une paire de clé publique/privée liées mathématiquement. La clé publique sert à chiffrer le message. Elle se calcule à partir de la clé privée de Bob. Dans ce schéma, Alice va chiffrer avec la clé publique de Bob qui déchiffrera avec sa clé privée, ce qui assure à Bob d'être le seul à pouvoir accéder au contenu. Cette solution a pour avantage de ne pas avoir à pré échanger une clé comme pour le chiffrement symétrique. Il faut néanmoins s'assurer qu'Alice a bien récupérée la clé publique de Bob. Des algorithmes de distributions de clés sont alors nécessaires. Dans le cas des clés asymétriques, l'infrastructure Public Key Infrastructure (PKI) est utilisée. Elle définie un ensemble de procédures et de rôles permettant de créer, gérer et distribuer des certificats numériques. Ces certificats sont ensuite utilisés pour générer des clés publiques mais également permettre à une autorité de lier identité et clés publiques. Un nœud souhaitant obtenir la clé publique d'un voisin va interroger cette autorité de confiance.

L'algorithme Rivest Shamir Adleman (RSA) fait partie de cette catégorie de cryptographie asymétrique. Des solutions à base d'Elliptic Curve Cryptography (ECC) commencent également à être déployées. Elles permettent d'obtenir un niveau de sécurité équivalent au RSA avec des clés de tailles bien inférieures. Ainsi, pour assurer un niveau de sécurité équivalent, les clés utilisées par ECC font 163 bits contre 1024 bits requis avec l'algorithme RSA ce qui est un avantage dans un environnement contraint.

Dans les WSN, les nœuds sont contraints en énergie et en mémoire. La cryptographie symétrique étant plus rapide la consommation d'énergie est donc plus faible. De plus, pour un même niveau de sécurité, la taille des clés pour un chiffrement symétrique est inférieure à celle d'un chiffrement asymétrique. La solution de cryptographie symétrique est donc privilégiée dans les WSN.

Néanmoins, l'une des difficultés du déploiement de la sécurité dans les WSN concerne le choix des algorithmes utilisés pour le chiffrement. Les outils classiques de cryptographie ne sont pas forcément adaptés aux besoins et contraintes de ces nouveaux réseaux de capteurs. La cryptographie légère a donc été développée afin de permettre la conception et l'analyse d'outils de sécurité dédiés. Son but est de fournir des mécanismes de sécurité fonctionnant à des coûts très faibles de manière à être embarqués dans des environnements contraints. Les algorithmes utilisés doivent être étudiés vis-à-vis de la taille de leur code et des données, le temps de calcul et la consommation d'énergie nécessaire.

L'une des contraintes concerne notamment la taille des clés utilisées. En effet, la quantité de mémoire RAM stockant le bloc à chiffrer et la clé limite la taille maximale de ces éléments. Afin d'assurer un niveau de sécurité convenable et une empreinte mémoire limitée, des clés de 128 bits seront préférées pour le chiffrement ou l'authentification. L'implémentation logicielle de ces nouveaux algorithmes doit également faire appel à des opérations logiques ou arithmétiques élémentaires comme la fonction XOR ou les fonctions de décalage.

Des AES légers ont été publiés pour être embarqués dans des plateformes contraintes comme Nettle AES [18] ou encore Tiny-AES [19].

2.3.2 L'authentification du message et l'intégrité des données

Afin d'assurer l'authentification de la source du message ainsi que l'intégrité des données, un code d'authentification de message ou Message Authentication Code (MAC) est utilisé.

Le MAC est généré grâce à des fonctions de hachage et d'une clé secrète.

Les fonctions de hachage sont des fonctions de compression h dont l'entrée est une suite binaire de taille arbitraire et qui génèrent une sortie appelée haché de taille fixe n. La fonction h est à sens unique ce qui implique que pour un message m, le haché $h(m)$ obtenu est rapide à calculer mais difficile à inverser. Ainsi, un attaquant connaissant la fonction h utilisée et la sortie obtenue $h(m)$ ne pourra remonter au message m. Le haché permet de vérifier que le message n'a pas subi de modification.

Le calcul du MAC prend en entrée le message e et la clé symétrique privée k et produit un $tag = mac_e = h_{k,e}$. L'émetteur de la trame envoie alors $e||mac_e$, où $||$ représente la concaténation. Le récepteur possédant la même clé pourra calculer le haché du message r reçu lié à la clé ($mac_r = h_{k,r}$) et le comparer avec le MAC reçu (mac_e). L'égalité $mac_r = mac_e$ prouve l'intégrité du message et l'authentification de l'émetteur.

Néanmoins, l'utilisation du MAC ne permet pas de distinction entre des nœuds partageant la même clé. De nombreux algorithmes existent comme HMAC-SHA-1 ou HMAC-MD5.

2.3.3 La distribution des clés

Les clés utilisées dans un WSN peuvent être partagées de trois façons :

- commune à tout le réseau. C'est une clé globale connue de tous les nœuds. Cette solution est la plus simple et la plus économique en terme de mémoire. Néanmoins, le niveau d'authentification est faible et un nœud compromis rend toutes les communications accessibles à un attaquant.
- clé de groupe. Un nombre limité de nœud du réseau partage la même clé. Ces nœuds sont choisis suivant des caractéristiques communes (localisation, rôle...). Plusieurs groupes sont définis. On utilise alors une clé différente pour communiquer entre chaque groupe. Cette technique réduit l'impact d'une compromission à un groupe de nœuds.
- partagée par paire de nœud. Chaque nœud possède une clé différente pour communiquer avec son voisin. Cette solution est coûteuse en mémoire mais permet de limiter l'impact de la compromission d'un nœud.

Indépendamment du type de clé, la distribution des clés symétriques partagées posent problème pour des réseaux dont le déploiement peut être aléatoire, vaste et varier rapidement.

Dans une méthode centralisée, une unique partie de confiance crée et distribue les clés utilisées dans le réseau. Assurer ces services tout en gardant de bonnes performances réseaux est alors compliqué. Un nœud doit pouvoir communiquer avec cette entité rapidement afin d'obtenir la clé. Or, la demande peut se perdre et le protocole ne pas aboutir. De plus, le nœud doit être légitimé auprès de l'entité afin d'obtenir la clé.

Dans une méthode décentralisée ou distribuée, il n'existe pas d'entité pour la gestion des clés. Ce sont les nœuds qui organisent le partage et la création. Une solution consiste à stocker dans la mémoire, lors de la fabrication des nœuds, la ou les clés nécessaires aux communications. Les clés symétriques peuvent être dérivées d'une clé maitre stockée. Eschenauer et al. dans [20] proposent un mécanisme probabiliste où chaque nœud possède un sous ensemble des clés déployées dans le réseau. Ces sous ensembles sont créés de manière à ce que la probabilité que deux nœuds possèdent au moins une clé commune soit élevée. Cette solution nécessite que le constructeur déploie les clés. Ces clés par défaut seront alors communes à tous les nœuds d'un même fabriquant. Or, deux acheteurs auront donc la même clé par défaut, ce qui représente une faille de sécurité.

Une autre solution consiste à définir les clés lors du déploiement. Le propriétaire du réseau devra alors définir la ou les clés pour chaque nœud. Ce travail peut être fastidieux pour des réseaux vastes et compliqués à réaliser pour des utilisateurs lambda.

Elles peuvent également être obtenue grâce à des métriques communes aux nœuds. L'utilisateur va alors déployer son réseau sans se soucier des clés. Celles-ci sont obtenues par la suite. Delaveau, Mueller et al.

dans [21] proposent de générer les clés à partir des aléas du lien radio et grâce à la réciprocité du canal entre deux nœuds. Dans ce schéma, chaque paire de nœuds possède une clé différente pour communiquer. Or, dans des réseaux vastes, il est compliqué de stocker une clé par lien et le choix se porte sur une clé de groupe.

La distribution des clés est un mécanisme non trivial et représente aujourd'hui un challenge dans les WSN.

2.4 Les attaques contre la sécurité dans les WSN

Hennebert et al. dans [22] expliquent qu'assurer la confidentialité ainsi que mettre en place les autres mécanismes de sécurité dans les WSN n'est pas simple. Dans [22], les auteurs expliquent les difficultés qu'il existe pour l'établissement de la sécurité dans l'IoT, pour la distribution et la gestion des clés au sein du réseau et les bonnes pratiques à appliquer pour sécuriser au mieux un réseau.

Les communications sans fil ont lieu dans des réseaux sans réelle infrastructure entre nœuds contraints s'organisant dans des topologies aléatoires ce qui rend les solutions de sécurité classiques inadaptées. La nature de ces réseaux offre de nouvelles vulnérabilités exploitables comme un média ouvert simplifiant l'injection de données. La cryptographie ne permet pas de résoudre tous les problèmes de sécurité des WSN.

2.4.1 Les modèles d'attaquant

Afin de fournir une contre mesure adaptée, le premier critère à prendre en compte est l'objectif ou intention de l'attaquant. Ce critère permet de définir quel est le but recherché par un attaquant lorsqu'il lance son attaque.

Quatre grandes intentions peuvent être définies en accord avec celles retenues par Znaidi dans [23] :

- L'interruption des communications. Les paquets n'atteignent pas leurs destinations ce qui peut impacter la qualité de service. L'attaquant peut vouloir modifier les paquets ou provoquer leurs pertes. Si les paquets sont des paquets de routage cela peut entrainer des topologies non optimisées voire la destruction des liens entre les nœuds.
- Exhaustion. Les nœuds étant contraints, l'attaquant veut épuiser leurs ressources comme l'énergie. S'il parvient à cibler un nœud important du réseau comme le puits ou le nœud racine alors il peut entrainer la destruction complète du réseau.
- Identification. L'attaquant souhaite intégrer le réseau et être reconnu comme nœud légitime. L'absence de lien physique facilite les attaques par usurpation d'identités. Ainsi un attaquant peut vouloir prétendre être un nœud légitime en lui volant son identité.
- L'autorisation. Ces attaques ont pour but de contrer les mécanismes de contrôle d'accès. L'attaquant souhaite pouvoir accéder à des ressources comme les clés de chiffrement. Il peut vouloir extraire les données contenues dans un nœud existant en le compromettant.

Le statut de l'attaquant vis-à-vis des autres nœuds du réseau lui offre des capacités plus ou moins importantes. Dans le cas d'un attaquant interne, celui-ci à préalablement corrompu un nœud légitime. Il possède donc à présent le même matériel cryptographique que les nœuds du réseau et se manifeste comme légitime. Il est autorisé à accéder aux ressources. Dans le cas d'un attaquant externe, celui-ci est considéré comme étranger par les nœuds légitimes. Il ne connait pas les secrets du réseau et n'est pas autorisé à accéder aux ressources.

Deux attaquants peuvent vouloir atteindre le même but sans toutefois déployer/posséder les mêmes moyens. Dans le premier cas, un attaquant est dit fort s'il possède des caractéristiques supérieures aux nœuds contraints du réseau (antenne plus puissante, alimentation...). Ce type d'attaquant peut utiliser un PC portable. Il n'est également pas obligé de respecter les réglementations en termes de puissance d'émission. Si l'attaquant ne déploie pas de moyens supérieurs aux autre nœuds du réseau alors celui-ci est appelé attaquant ordinaire. L'attaque peut être menée avec le même type de nœud que ceux déployés ce qui lui permet de "se fondre" plus facilement. Un individu lambda peut mener l'attaque avec du matériel facilement accessible et pour autant être critique pour le réseau et son fonctionnement.

2.4.2 Les attaques contre la sécurité

Dans cette section, je vais présenter certaines attaques traditionnelles dans les WSN classées suivant la couche du modèle OSI (cf. Figure 2.1) où elles agissent. Cette classification est basée sur celle réalisée dans [24] par Sen. L'auteur décrit les vulnérabilités existantes par couche dans les WSN ainsi que les solutions existantes de l'état de l'art. Il fait également un travail similaire sur les Cognitive WSN, réseaux où les nœuds sont capables d'adapter leurs couches Physiques à leur environnement afin d'obtenir des communications plus fiables et efficaces.

Table 2.1 – Attaques de sécurité par couche OSI.

Couches	Attaques
Physique	Brouillage
MAC	Epuisement
	Collision
Réseau	Selective Forwarding / Blackhole
	Sinkhole / Hello flood
	Sybil
	Wormhole
Application	Inondation
	Désynchronisation

2.4.2.1 Attaques sur la couche Physique

Le brouillage est le moyen le plus simple de réaliser du déni de service (Denial of Service (DoS)) sur la couche Physique. Il est extrêmement critique pour un réseau.
Dans cette attaque, l'adversaire exploite le caractère libre du média afin de perturber le réseau. Il tente d'interférer avec les fréquences radio utilisées afin de nuire aux communications. Cette attaque est simple et efficace notamment à cause de l'utilisation d'une fréquence unique.
Zheng et al. dans [25] expliquent que l'attaque peut être menée en émettant continuellement un signal radio sur la même fréquence que celle utilisée par les nœuds légitimes. Une autre solution consiste à n'émettre qu'à intervalles réguliers. Enfin, l'attaquant peut choisir de n'émettre que s'il détecte un signal sur le canal. Dans ce cas, l'attaquant devra être plus intelligent et posséder des techniques de détection de signal.
Cette attaque a donc besoin de capacités techniques supérieures afin de combler l'apport en énergie nécessaire pour l'émission continue. Toutefois, un nœud avec des capacités ordinaires pourra tenter de mener cette attaque sur un nœud stratégique comme la racine. Il sera ainsi capable d'interrompre les communications du réseau entier. Pour mener cette attaque, il n'est pas nécessaire d'appartenir au réseau. Un attaquant externe peut lancer efficacement cette attaque. L'objectif d'un attaquant effectuant du brouillage est donc de perturber les communications et de nuire à la qualité de service.

2.4.2.2 Attaques sur la couche MAC

Sur la couche MAC, une attaque par collisions peut être menée. Une collision est définie quand deux nœuds tentent de communiquer simultanément, intentionnellement ou non, sur la même fréquence. L'attaquant va alors envoyer son propre signal afin de créer des collisions avec les nœuds légitimes. Ces interférences ont pour but de réaliser une attaque DoS consommant moins d'énergie que celle réalisée à la couche Physique.
Lorsqu'une collision arrive, le message ou une partie de celui-ci est modifié. Cela va provoquer une erreur dans la somme de contrôle (*checksum*). En effet, afin de détecter les erreurs de transmissions, un *checksum* est calculé vis-à-vis des bits de la trame. Il est ajouté à la trame avant émission. Ce *checksum* est calculé de nouveau à la réception et comparé à celui contenu dans la trame envoyée. Si un bit diffère, alors le *checksum* ne sera pas égal et le message sera supprimé entrainant une nouvelle émission.
L'attaquant tente de cibler certains types de message comme les messages d'acquittement. Les nœuds légitimes

ne recevant pas l'acquittement, l'envoi de la donnée est répété. Ces collisions peuvent également être menées grâce à la connaissance des protocoles utilisés pour l'accès au média.
Par exemple, Sajjad et al. dans [26] décrivent comment le standard définis des slots de communications attribués à chaque nœud. Ils expliquent comment il est possible d'utiliser ce mécanisme pour réaliser des collisions et du déni de service.
En effet, lorsqu'un nœud souhaite communiquer de façon certaine et sans interférences avec un autre nœud, il demande une attribution de slot. Ce protocole consiste en un échange de trames avec un coordinateur qui connait les slots déjà attribués et alloue les restants. Un attaquant écoutant le protocole peut récupérer le slot alloué et lors de la prochaine communication émettre au même moment. Il réalisera alors une collision. Le but d'un tel attaquant est d'interrompre les communications mais également d'épuiser les ressources. Cette attaque peut être menée par un attaquant ordinaire externe au réseau.

L'épuisement est une autre attaque de la couche MAC. Du fait que les WSN soient basés sur des nœuds contraints, les attaques par épuisement des ressources sont très efficaces.
Une de ces attaques consiste à perturber le canal en demandant continuellement son accès. Le canal apparait alors comme occupé tout le temps. Les nœuds légitimes ne peuvent alors émettre et vont épuiser leur batterie pour accéder à celui-ci. Cette attaque est appelée *Backoff manipulation*.
Comme l'expliquent Ghildiyal, Mishra et al. dans l'étude [27] sur les attaques par déni de service dans les WSN, l'une des méthodes est d'émettre un nombre important de trames Request To Send (RTS), trame appartenant au protocole permettant à un nœud de demander l'accès au canal. Ainsi, les nœuds légitimes vont demander l'accès au média sans jamais obtenir de réponse positive. Les nœuds vont ainsi épuiser leurs batteries. Un attaquant peut également obliger un nœud à effectuer, de manière répétée, l'un des protocoles, par exemple les protocoles de maintien du routage, sans forcément aboutir. Il va alors débuter une procédure et en cours la stopper. Si les nœuds ont beaucoup de tâches à réaliser, il est alors possible d'épuiser les ressources disponibles.
Pour mener ce type d'attaque, des capacités ordinaires sont nécessaires. L'attaquant peut être externe. Le but étant le même que celui recherché pour une attaque collision.

2.4.2.3 Attaques sur la couche Réseau

La couche Réseau est sujette à de nombreuses attaques.

Dans l'attaque *selective forwarding*, le nœud malveillant étudie le trafic. Il sélectionne alors les messages qu'ils souhaitent router et ceux qu'il souhaite jeter. Comme l'expliquent Khan, Yang et al. dans leur étude [28] sur les attaques *selective forwarding*, plusieurs types existent. Dans un premier temps, l'attaquant peut choisir de sélectionner des messages vis-à-vis d'un nœud ou d'un groupe de nœud. Il peut également sélectionner les messages par rapport à leur nature. On parle alors d'attaque *Neglect* ou *Greed*. En négligeant de router les messages concernant le protocole de routage utilisé pour le maintien et le choix des routes, il peut ainsi modifier la topologie et la rendre moins optimale. Dans ce cas, les nœuds vont consommer plus d'énergie et épuiser plus rapidement leurs ressources. La qualité de service sera également impactée. En laissant passer les messages pour l'application, les nœuds du réseau pensent alors envoyer correctement leurs données et la détection de l'attaquant est compliquée.
Il peut également augmenter la priorité des messages le concernant au dépend de ceux des nœuds légitimes. Enfin, il peut ajouter du retard dans le routage semant la confusion au sein du réseau. Les nœuds observent alors des délais de routage supérieur à ceux estimés pour la topologie et en déduisent que cette topologie a changée.
L'attaque *Blackhole* est un exemple d'attaque *selective forwarding* où l'attaquant ne route aucun paquet. Un attaquant veut mener cette attaque dans le but d'effectuer un déni de service sur les communications et nuire à la qualité de service. Cette attaque peut être réalisée avec des nœuds ordinaires appartenant au réseau.

Dans une attaque *sinkhole* ou *hello flood*, le nœud malveillant tente d'attirer un maximum de trafic. Cette attaque est l'une des plus dangereuse pour les WSN. Comme l'expliquent Krontiris, Giannetsos et al. dans [29] plusieurs stratégies peuvent être utilisées pour mener à bien cette attaque. Ils démontrent dans leur article la faisabilité de cette attaque ainsi que le faible coût d'implémentation pour un attaquant. Le but est de placer un nœud malicieux proche du nœud puits afin d'attirer un maximum de trafic.

L'une des méthodologies possible consiste à déclarer à ses voisins posséder le meilleur chemin jusqu'à la station de base et donc les attirer pour être ses fils faisant passer ainsi par lui un maximum de trafic. C'est ce que l'on appelle l'attaque *hello flood*. L'attaquant utilise des trames de routage forgées incluant des métriques utilisées pour le calcul de la route optimale meilleures que celles de ces voisins. Les nœuds voisins vont le choisir pour être leur père. Plus il possède de fils et plus il est sûr d'observer un maximum de trafic. Si cette attaque est menée avant une attaque *selective forwarding*, cette dernière devient plus puissante et plus simple à réaliser. Grâce à cette attaque, un nœud malveillant interne au réseau peut produire une mauvaise topologie et donc nuire aux communications. S'il n'appartient pas encore au réseau, cette attaque va lui permettre d'essayer de s'identifier afin d'appartenir au réseau. L'attaquant doit posséder des capacités supérieures afin de leurrer les autres nœuds.

Un attaquant menant une attaque *sybil* va tenter de s'introduire dans le réseau. Pour cela, il va usurper l'identité de nœuds légitimes. Il va alors copier l'identité d'un ou plusieurs nœuds dans le même nœud physique ou dans plusieurs matériels afin d'avoir accès à une large partie du réseau et de participer aux communications. Il veut ainsi pouvoir surveiller le trafic voire influencer les données collectées.
Cette attaque est également intéressante pour contrer les mécanismes de défense utilisant le vote des nœuds du réseau. En effet, en possédant plusieurs identités, le nœud peut influencer le résultat du vote et éviter d'être mis sur liste noire. L'autre effet est que le nœud dont l'adresse a été usurpée n'a plus accès aux données qui lui étaient adressées au préalable.
Une attaque similaire consiste à insérer des nœuds répliqués dans le réseau afin de prendre part aux communications, et cela, à des points stratégiques du réseau. Il doit tout d'abord accéder aux données enregistrées dans un nœud en accédant physiquement à celui-ci et les répliquer. Les effets sur le réseau sont identiques à ceux observés lors d'une attaque *sybil*. Newsome, Shi et al. dans l'article [30] présentent une taxonomie des différentes attaques *sybil* possibles. Cette taxonomie leur permet de mieux comprendre les effets de l'attaque sur le réseau et donc d'adapter la contre mesure. Le but de cette attaque est de nuire à la qualité de service mais également d'usurper l'identité de nœud légitime afin de joindre le réseau. Un nœud classique peut mener cette attaque facilement sans déployer de capacité particulière. L'attaquant d'abord externe au réseau va devenir un nœud légitime grâce à cette attaque.

L'attaque *wormhole* consiste à créer un tunnel de communication hors bande entre deux attaquants éloignés physiquement afin de berner les nœuds légitimes. Cette attaque est dangereuse pour les protocoles de routage. Différents types d'attaques *wormhole* sont détaillés par Maidamwar et al. dans [31]. Les deux nœuds malicieux vont déployer un lien de plus faible latence par rapport aux autres liens du réseau. Pour cela, ils vont utiliser un lien filaire ou une communication sans fil longue portée et de plus forte puissance que les antennes des nœuds classiques. Les données vont être collectées par le premier attaquant dans une partie du réseau et transmises au deuxième nœud malveillant se situant dans l'autre partie. Le nœud malveillant se trouvant de l'autre côté du tunnel va alors transmettre les paquets aux nœuds voisins légitimes. Le but est de placer le premier attaquant proche du nœud puits et le second dans une partie éloignée. Le lien entre les deux attaquants étant meilleur et plus rapide que le lien sans fil, les nœuds voisins les incluent plus facilement dans leurs chemins. Ainsi, la topologie du réseau est modifiée et les nœuds pensent être à proximité, ce qui perturbe les communications. L'adversaire va mener cette attaque dans le but de créer une fausse topologie mais également de s'authentifier auprès du réseau. Il peut également lancer cette attaque dans le but de mener ensuite une attaque plus puissante comme les attaques par DoS. Afin d'obtenir un meilleur lien, l'attaquant doit posséder des capacités supérieures notamment au niveau de la radio ou du lien physique. Contrairement aux autres attaques, l'attaque *wormhole* nécessite le déploiement d'au moins deux nœuds dans le réseau. Néanmoins, les nœuds n'ont pas l'obligation d'être internes pour que l'attaque fonctionne.

2.4.2.4 Attaques sur la couche Application

Dans l'attaque inondation, un adversaire souhaite épuiser les ressources des nœuds. Il va lancer les protocoles de connexion de la couche Application sans en permettre l'aboutissement. En répétant ce procédé sur un nœud légitime, les ressources de ce dernier vont s'épuiser. Cette attaque peut être menée par un nœud ordinaire appartenant au réseau.

L'attaque désynchronisation a pour but de nuire aux communications et à la qualité de service. Un attaquant va tenter de leurrer deux nœuds connectés entre eux en leur faisant croire qu'ils sont désynchronisés. Il forge alors une trame qu'il envoie à chacune des deux parties indiquant qu'ils sont désynchronisés. Dès réception, les nœuds vont lancer une procédure de resynchronisation et rompre le lien actuel. Un attaquant interne peut mener cette attaque avec des capacités classiques.

2.4.3 Etat de l'art des contre mesures

De nombreuses études ont été menées afin de déterminer les mécanismes de sécurité les mieux adaptés aux WSN. Différentes approches peuvent être envisagées afin d'assurer la sécurité comme dans la mise en place même des protocoles ou par l'ajout de mécanismes extérieurs comme les Intrusion Detection System (IDS).

2.4.3.1 Contre mesures sur la couche Physique

Lorsque l'attaque DoS par brouillage est menée sur la couche Physique, la contre mesure classique consiste à utiliser des techniques d'étalement de spectre. Le *Frequency-Hopping Spread Spectrum* (FHSS) ou étalement de spectre par saut de fréquences consiste à transmettre le signal en changeant rapidement de canal de fréquence. Les sauts suivent une séquence pseudo aléatoire connue de la source et de la destination. Ainsi, un attaquant ne connaissant pas le canal d'émission devra polluer toutes les fréquences en même temps, ce qui est complexe à mettre en place. Néanmoins, cette technique est très onéreuse en complexité et en coût. Les nœuds de capteurs actuels ne sont pour la plupart capables d'émettre que sur une seule fréquence. Une autre méthode présentée par Xu, Ma et al. dans [32] consiste à tolérer ces attaques et à identifier les nœuds ou la partie du réseau concernée par le brouillage. Ces nœuds sont ensuite isolés des communications.

2.4.3.2 Contre mesures sur la couche MAC

Les codes correcteurs d'erreurs [33] peuvent être utilisés pour prévenir les attaques par DoS sur la couche MAC. Cette solution est efficace lorsque l'erreur touche un faible nombre d'octets mais engendre des temps de latence et un surcoût d'énergie. Dans le cas où l'attaquant utilise les protocoles et le standard pour mener une attaque plus puissante, l'utilisation du chiffrement permet de cacher l'information du slot alloué pour la communication et oblige l'attaquant à effectuer une attaque collision avec détection en temps réel des communications.

Lorsqu'à la couche MAC, l'attaquant tente d'épuiser les ressources du nœud, une contre mesure consiste à mettre en place un taux maximal de paquets de contrôle autorisé pour chaque nœud (quota). Ainsi, les demandes trop fréquentes ou nombreuses seront rejetées. Néanmoins, ce quota ne peut être défini en dessous du débit maximum du réseau. L'utilisation de méthodes de division temporelle [34] pour l'accès au canal permet également de limiter le temps que possède un nœud pour communiquer, réduisant le temps que possède un attaquant pour lancer son attaque.

2.4.3.3 Contre mesures sur la couche Réseau

Les attaques *selective forwarding* et *blackhole* peuvent être contrées par des mécanismes de routage multi chemins dynamiques tel que celui décrit par Ganesan et al. dans l'article [35]. Dans ces schémas, le prochain saut dans le chemin de routage est choisi aléatoirement, ce qui réduit le risque que le message soit acheminé via un attaquant mais augmente la consommation d'énergie. Des mécanismes d'authentifications peuvent être déployés afin de permettre à un fils de légitimer son père et donc le prochain saut dans le chemin de routage. Une autre contre mesure est de permettre au nœud du réseau ou à des nœuds surveillants (*watchdog*) de vérifier le bon routage des trames. Xin-Sheng, Yong-zhao et al dans [36] proposent un système de surveillance léger distribué pouvant être embarqué dans les nœuds du réseau. Les voisins observent la transmission des paquets. Si un paquet n'est pas transmis, le voisin ayant observé le problème se charge de

router le message en concordance avec le schéma de routage. Il émet également une alarme à ses voisins afin d'avertir de la localisation d'un potentiel attaquant. Le nœud est alors exclut. Les auteurs de [36] ont choisis des contraintes fortes pour le modèle du réseau qui ne collent pas forcément avec la réalité des WSN. Si des nœuds sont en mouvement, le schéma n'est plus aussi performant. Khan, Yang et al. dans [28] donnent un aperçu des différentes méthodologies pour la détection des attaques *selective forwarding* ainsi que les inconvénients de chacune. Les auteurs dressent un tableau comparatif permettant notamment de séparer les solutions centralisées des décentralisées.

Dans une attaque *sinkhole*, l'attaquant utilise les métriques de routage. Chiffrer les trames contenant la métrique utilisée pour le choix du chemin rend impossible cette attaque sans une connaissance préalable du matériel de sécurité. Cette solution est donc efficace uniquement si le nœud ne peut obtenir l'accès au réseau par une autre attaque préalable (attaquant externe). Une autre solution consiste à choisir une métrique non fournie par les voisins ou une métrique réciproque aux deux nœuds et liée au lien radio comme la puissance en réception du signal reçu (Received Signal Strength Indication (RSSI)). Il est possible de déployer des protocoles de routage qui vérifient la fiabilité bidirectionnelle du chemin comme la solution proposée par Karlof et al. dans [37] grâce à des acquittements *end-to-end* contenant des informations sur la latence et la qualité. Enfin, Coppolino, D'Antonio et al. dans [38] proposent de permettre à chaque nœud du réseau de détecter si un nœud est suspect. Dès qu'une alarme est levée concernant un nœud, son identité est ensuite inscrite dans une liste noire. C'est alors un agent central qui va prendre la dernière décision concernant le nœud identifié comme malveillant et en informer les nœuds du réseau afin de le retirer des communications.

L'attaque *sybil* repose sur l'utilisation d'identifiants existants usurpés par un attaquant. Permettre la mise en place de l'authentification est la première contre mesure à prendre. Contre les attaques par réplication de nœud, des solutions similaires peuvent être déployées. Newsome, Shi et al. dans [30] présentent quatre solutions pour valider les identités adaptées aux différentes attaques *sybil* identifiées. Une contre mesure est de garder le nombre d'instance de chaque identité ou de faire stocker une information concernant la localisation des nœuds par une autorité de confiance ou dans des Distributed Hash Table (DHT). Dans la solution de vérification de la localisation par la station de base, chaque nœud envoie une liste de ses voisins et de leurs localisations. La station de base vérifie qu'il n'existe pas deux localisations pour le même nœud. Cette solution peut s'avérer compliquée pour un réseau dense.

Un attaquant menant une attaque *wormhole* n'a pas besoin de comprendre ce qu'il transmet. Le chiffrement est donc inefficace face à cette attaque. L'attaque *wormhole* est difficile à détecter. Znaidi dans [23] détaille les différentes contre mesures existantes mais également les inconvénients de chaque solution. Il est possible d'utiliser des antennes directionnelles comme celles proposer par Hu et al. dans [39] pour réduire la possibilité de mener une telle attaque. Cette technique permet de déployer un système de localisation moins cher que d'autres techniques comme le GPS mais est sujette à de nombreuses pertes de connectivité dans des réseaux trop denses. Dans sa solution, Znaidi propose de permettre la détection de l'attaque par chaque nœud du réseau grâce aux informations locales des nœuds. Ils utilisent pour cela la liste des voisins 1 et 2 sauts. Cette solution repose sur le postulat que si deux nœuds sont voisins alors ils possèdent au moins un nœud voisin en commun.

2.4.3.4 Contre mesures sur la couche Application

Afin d'éviter les attaques inondations de la couche Application, l'une des contre mesures traditionnelles est l'utilisation de challenge ou de "client puzzle". Proposée dans [40] par Aura, Nikander et al., cette preuve de travail permet de démontrer à son voisin sa connexion. Ainsi, pour qu'une connexion soit effective, chaque nœud doit prouver celle-ci en résolvant le challenge fourni par un serveur avant de pouvoir accéder aux ressources. Ces énigmes représentent une complexité mathématique limitant les demandes d'un attaquant.

Contre la désynchronisation, l'authentification de tous les paquets doit être renforcée. L'identité et l'intégrité peuvent être ainsi vérifiées.

De nombreuses attaques pouvant être menées aux différentes couches du modèle OSI sont facilitées par les ressources limitées des WSN. Des mécanismes de protection sont toutefois disponibles. Néanmoins, afin de déclencher le mécanisme approprié, la première étape consiste à mettre en place des outils de détection permettant de classifier et détecter l'attaque.

2.4.3.5 Mesures de détection

Une solution possible est alors de déployer un IDS dans le WSN. Ce mécanisme a pour but de repérer les activités anormales ou suspectes et de lancer en conséquence les contre mesures adaptées. L'IDS va alors détecter soit un comportement inhabituel soit un scénario inhabituel. Abduvaliyev, Pathan et al. dans l'article [41] classifient les IDS selon la technique de détection utilisée. Il peut détecter une signature, une anomalie ou être basé sur des spécifications.

Détection d'une signature.

Dans ces IDS un attaquant est détecté grâce à des motifs d'attaques connus. La démarche est similaire à celle utilisée pour les antivirus. Elle consiste à rechercher dans l'activité des nœuds, des signatures ou empreintes recensées et connues. Une bibliothèque de signatures est donc créée lors d'une première phase par l'administrateur du réseau. L'activité en temps réelle est ensuite comparée à cette base de signatures afin de détecter un comportement suspect. Cette technique, très utilisée dans les IDS classiques, est moins convenable pour les WSN. En effet, cette méthode nécessite la connaissance et le stockage des différents motifs mais également un coût de calcul important pour les algorithmes de comparaison. De plus, cet IDS manque de flexibilité dans l'ajout de nouveaux motifs et nécessite des mises à jour fréquentes. Ces IDS sont très efficaces s'ils connaissent l'attaque mais inefficace sinon. Dans la plupart des cas, ce type d'IDS n'est pas utilisé. Une approche similaire appelée *watchdog* est plutôt envisagée. Dans cette technique, la nature sans fil est exploitée afin de permettre aux voisins d'un nœud récepteur de jouer le rôle d'IDS. Ainsi, même si une communication ne lui est pas adressée, un voisin peut surveiller celle-ci et observer son bon comportement. Néanmoins, cette solution nécessite que les nœuds écoutent tout le temps et doivent également analyser les paquets reçus ce qui consomme beaucoup d'énergie. Dans leur article [42], Amin, Siddiqui et al. proposent un IDS réduisant les coûts de stockage ainsi que l'énergie nécessaire à l'envoi de l'alerte au nœud racine grâce à l'utilisation de filtres de Bloom pour le stockage des signatures. Les filtres de Bloom [43] sont des structures de données qui permettent d'optimiser l'espace de stockage. Créée en 1970, cette structure utilise les fonctions de hachage. Un filtre va hacher une donnée et stocker son résultat. Ainsi l'espace de stockage nécessaire est réduit. Un utilisateur pourra uniquement interroger la structure pour savoir si la donnée est enregistrée ou non. Néanmoins, de par son fonctionnement, le filtre de Bloom est sujet à de nombreux faux positifs. Dans la solution de [42], un paquet reçu va d'abord être passé dans un filtre de Bloom afin de déterminer si celui-ci correspond à une des signatures stockées. Si c'est le cas, un message est envoyé au nœud racine contenant le code généré de la signature.

Détection d'anomalies.

Dans ce type d'IDS, les évènements sont analysés afin d'identifier une utilisation non autorisée. Une première phase consiste à identifier ce qu'est un comportement normal et qui servira de référence pour détecter l'anomalie. Pour cela, le réseau de référence est analysé. Il est supposé que pendant cette phase aucun attaquant n'est présent. Les données utilisées pour la décision peuvent être des données statistiques (durée retransmission, nombre de paquets...). Ce type d'IDS permet de détecter, contrairement au précédent, de nouvelles attaques plus efficacement mais possède un taux de faux positifs important. Un faux positif est décrit comme la détection d'un nœud normal comme malveillant. De plus, une phase d'apprentissage est nécessaire ce qui est très coûteux dans des réseaux contraints. L'étude [44] menée par Xie, Han et al. explique que la recherche de solutions permettant de détecter des anomalies dans les WSN représente encore de gros challenges (2011). Ils font un inventaire de certaines solutions existantes et discutent des avantages et inconvénients de chacune. Aucune des solutions présentées ne satisfait tous les besoins et de nombreuses améliorations sont nécessaires avant la mise en place dans les WSN.

Détection basée sur des spécifications.

Cet IDS est similaire au précédent si ce n'est que la phase d'apprentissage est remplacée par une définition manuelle de ce qu'est un comportement normal. Cette amélioration permet de diminuer le taux de faux positifs. Ces IDS tentent de prendre le meilleur des deux IDS précédents. Néanmoins, comme pour le premier IDS, un administrateur doit définir les spécifications, ce qui est difficile et chronophage. Plusieurs approches sont détaillées par Abduvaliyev, Pathan et al. dans [41]. Les IDS peuvent être embarqués dans les nœuds du réseau. C'est cette méthode qu'utilisent Krontiris, Dimitriou et al. dans [45] pour détecter l'attaque *sinkhole*. Les mêmes auteurs dans [46] proposent LIDeA (LIghtweight Detection Architecture), un IDS léger

distribué. L'implémentation est testée dans un Operating System (OS) léger afin de détecter également l'attaque *sinkhole*.

Malheureusement, indépendamment du type d'IDS, toutes ces solutions sont pour l'instant peu développées et représentent encore un enjeu dans leurs réalisations et leurs mises en place. En effet, les IDS existants pour les réseaux IP classiques sont soit trop gourmands, soit prennent trop de place pour être embarqués dans un WSN.

Certains IDS dédiés WSN commencent à voir le jour comme ceux donnés précédemment mais ils subsistent des inconvénients à leur mise en place. De plus, suivant leur déploiement, les contraintes imposées par les nœuds obligent les IDS à ne contrer qu'une seule attaque ou qu'un seul type d'attaque. En effet, les IDS peuvent être localisés à plusieurs endroits du réseau.
Ils peuvent premièrement être embarqués dans une autorité de confiance qui fera tourner l'IDS et pourra, par exemple, être moins contrainte que le reste du réseau. Dans ce cas, l'IDS peut être complet et contrer plusieurs attaques. On parle d'IDS centralisé. Néanmoins, dans ce fonctionnement, il est nécessaire que l'autorité de contrôle puisse avoir accès à toutes les communications du réseau, ce qui est compliqué dans des réseaux vastes. De plus, si celle-ci venait à être compromise, la sécurité assurée par l'IDS serait caduque.
Une autre solution consiste à décentraliser l'IDS en l'embarquant dans les différents nœuds. Néanmoins, cela est coûteux en calcul et en énergie. De plus, les nœuds étant très contraints en mémoire, ils sont limités sur le nombre de règles ou de signatures à embarquer. Les IDS utilisent alors la même technique de détection sur différentes attaques. Tant que les attaques que subit le réseau ont un comportement identique (modification du nombre de paquets, chemin non optimal...) ces IDS peuvent détecter l'attaquant. En revanche, si une attaque avec un comportement complètement différent est mise en place, celui-ci est inutilisable.

Les exemples d'attaques du paragraphe précédent montrent qu'il en existe un grand nombre disponible et il faudrait alors déployer un IDS pour chacun. Le nombre d'IDS à déployer serait donc trop important. Dans l'étude [47], Darra et al. tentent de dresser une liste non exhaustive des attaques possibles sur les WSN ainsi que les IDS existant. Ceux-ci sont souvent efficaces contre les attaques de sécurité car elles agissent sur le réseau et peuvent être détectées. En revanche, les IDS sont bien souvent inutiles face aux attaques sur la vie privée.

2.5 La protection de la vie privée

Lors des communications, le réseau ne devrait pas indiquer les identités ainsi que les positions des nœuds le composant. Il ne devrait pas fournir de métadonnées sur son fonctionnement. Assurer la protection de la vie privée revient à déployer une solution permettant d'éviter la collecte massive et l'utilisation de ces métadonnées par un attaquant. Ces solutions doivent permettre le partage des données collectées par les capteurs tout en protégeant les informations sensibles. Il faut également que l'introduction d'un mécanisme de protection de la vie privée ne nuise pas aux mécanismes de sécurité.

2.5.1 Les modèles d'attaquant

Trois modèles sont considérés lorsque l'on parle de protection de la vie privée :

- Attaquant Externe. Cette attaque est menée par un nœud non autorisé extérieur au réseau. Il ne participe pas aux communications ni au routage et n'interagi pas avec les nœuds. Cette attaque est lancée par n'importe quel étranger au réseau souhaitant collecter un maximum d'information. Le but de l'attaque est de placer une antenne dans le réseau et d'intercepter le trafic à portée.
- Attaquant Interne. L'attaquant est capable de prendre le contrôle d'un nœud ou d'un équipement du réseau. Ainsi, il est perçu comme légitime et peut accéder aux ressources du réseau. Il possède également le matériel cryptographique nécessaire (clés, authentifiants...) pour participer aux communications. Il a alors accès à tout le trafic passant par lui. Néanmoins sa vision du réseau reste limitée. Son but est donc d'être positionné à des lieux stratégiques de manière à surveiller un maximum de trafic afin d'obtenir le plus d'information sur le réseau et ses nœuds.

- Attaquant Global. C'est un attaquant interne mais avec une vision complète du réseau. Il est donc plus puissant qu'un simple attaquant interne. Il est capable de tout contrôler et d'observer toutes les communications ayant lieu dans le réseau. Il peut s'agir de l'administrateur du réseau. Dans ce cas, un utilisateur souhaitera éviter que ce dernier ait accès à des informations personnelles le concernant. Le but de l'attaquant est de collecter toutes les informations disponibles dans le réseau complet.

2.5.2 Les attaques

La première attaque est l'écoute passive (*eavesdropping*). L'attaquant va écouter, de manière prolongée, les communications qui ont lieux dans un WSN. Il va collecter les paquets sur une ou plusieurs cibles spécifiques. L'écoute se fait via un intercepteur qui récupère, sans filtrer, toutes les trames au niveau MAC (trames brutes) et les enregistre dans un fichier. Ce fichier, soit pendant soit après la collecte, est analysé par un analyseur de réseau tel que Wireshark [48] qui permet de disséquer les différents champs des trames reçues. Les informations vont ensuite être extraites et exploitées afin de découvrir les identités des nœuds (adresses MAC, IP, ports UDP ...) mais également les caractéristiques du réseau. Cette attaque a pour but de découvrir le contenu des communications. Elle est menée par un attaquant externe au réseau. Cette attaque est souvent la première phase d'une stratégie d'attaque de sécurité.

La deuxième attaque permet de faire de l'analyse de trafic. L'attaquant externe doit alors chercher à déterminer, a posteriori, à partir des données collectées par écoute passive, des informations sur la localisation ou l'identification d'un nœud spécial (par exemple le nœud racine) ou des informations de fréquences d'échanges de trames, de motifs ou de topologie. Le but de cette attaque est de déterminer les vulnérabilités du réseau exploitables afin de mener des attaques de sécurité plus puissantes et ciblées.

Dans l'article [49], Luo, Ji et al. expliquent que l'analyse de trafic peut prendre trois formes.
L'une des premières informations disponibles par analyse de trafic concerne les données temporelles. Dans cette attaque, la donnée intéressante concerne le temps de routage d'une trame ou le temps nécessaire à une donnée pour aller de la source vers la destination (bien souvent le nœud puits). L'attaquant tente de retracer le chemin qu'a parcouru la donnée afin d'atteindre la localisation de la source. Dans ce type d'analyse, un attaquant connaissant le réseau, sa topologie et son fonctionnement, va pouvoir déduire du temps de transmission des paquets au nœud puits des informations de vie privée. Il va pouvoir, lors de l'émission d'une trame, déterminer la localisation du nœud source. Dans la littérature, cet exemple est appelé *problème du panda et du chasseur*. Il est utilisé pour illustrer les fuites d'informations lors de communications sans fil et leurs conséquences. En effet, dans cet exemple, un WSN est déployé afin de surveiller les déplacements des pandas dans une forêt de bambou. Ces derniers portent sur eux un tag les identifiant. Lorsque ceux-ci se trouvent dans une zone, les capteurs à proximité, envoient au nœud puits l'information. Si, au même moment, un chasseur par analyse de trafic arrive à retrouver les capteurs sources, il peut alors retrouver la localisation du panda et chasser plus efficacement. Ce problème existe également dans les domaines militaire ou industriel et peut avoir de lourdes conséquences. Dans ce type d'attaque, l'attaquant est externe et a pour but de localiser les sources des données collectées. Cela lui permet d'identifier les capteurs porteurs d'informations et donc de mener par la suite une attaque DoS sur ces cibles stratégiques. La qualité de service sera alors impactée. Néanmoins, il a besoin d'une vision de l'ensemble du réseau et de ses communications afin de retrouver la source. Pour ce faire, suivant la nature et la taille du réseau, il peut soit déployer un seul intercepteur avec une portée importante (cas d'un réseau de taille raisonnable) ou dans le cas d'un réseau vaste comme celui qui surveille l'habitat des pandas, déployer plusieurs intercepteurs couvrant toute la zone. Ainsi, même si l'observation du réseau complet nécessite l'achat de 1000 nœuds à 25$ chacun, la revente d'un seul panda peut rapporter 66 500$ en Chine (prix en 2003).

Dans la deuxième forme d'analyse de trafic, ce sont les données statistiques qui sont importantes et notamment le taux de paquets. Dans ce schéma, l'attaquant part du constat que les nœuds proches du puits routent plus de trames que des nœuds éloignés. Il peut ainsi tenter de localiser le puits en cherchant les nœuds ayant le plus grand débit. De même que pour l'analyse temporelle, l'attaquant a pour but de mener une attaque DoS pour nuire aux communications. L'attaquant va alors utiliser une phase d'apprentissage afin de connaitre le débit classique dû à l'application et aux protocoles réseaux. Deng, Han et al. dans l'article [50] donnent les figures représentant le taux de paquets par rapport à la position dans le réseau que peut obtenir

un attaquant par ce type d'attaque. Ils montrent qu'il peut très facilement localiser le nœud puits ou la station de base. Il peut ainsi se rapprocher de celle-ci.

Enfin, la dernière forme que peut prendre l'analyse de trafic consiste en l'utilisation des identifiants. Ces identifiants se trouvent dans les en-têtes de différentes couches. Les attaques permises, notamment dans les réseaux contraints, par ces identifiants sont de plus en plus étudiées. En 2016, l'Internet Engineering Task Force (IETF) publiait la RFC 7721 [51] sur les failles de sécurité et de vie privée des mécanismes de génération d'adresses existants.

Elle identifie quatre types d'attaques qui peuvent être menées grâce aux identifiants permanents :

- La corrélation d'activités. Tant qu'une adresse reste valide, même lorsqu'un nœud change de réseau, un attaquant peut associer les communications et donc les activités de cette adresse entre elles.
- La localisation. Un attaquant peut tenter de sonder un réseau à la recherche d'une adresse précédemment observée. Il peut ainsi retrouver la topologie d'un réseau sans fil ou observer les mouvements dans le réseau.
- Les attaques par scan d'adresse.
- L'exploitation des vulnérabilités liées aux spécificités d'un matériel. Les adresses MAC sont composées de 24-bits assignés par l'IEEE appelés Organizationally Unique Identifier (OUI). Ces bits identifient un fabriquant mais permettent également à un attaquant de connaitre les failles du matériel pour mener une attaque ciblée.

Grâce à de l'écoute passive, un attaquant peut donc déduire les liens entre les nœuds du réseau et donc connaitre la topologie du réseau. Il peut également utiliser ces identifiants afin de mener des attaques ciblées sur ces nœuds. Ces identifiants sont également importants car ils sont utilisés dans les systèmes de contrôle d'accès/authentification ou dans certains IDS afin d'identifier et de mettre sur liste blanche ou noire certains nœuds.

Les trois attaques d'analyse de trafic peuvent être combinées pour déduire plus rapidement et efficacement les informations de vie privée.

Une attaque *homing* peut également être menée grâce aux données collectées par écoute passive. L'attaquant va, dans un premier temps, effectuer une analyse de trafic afin de déterminer les nœuds importants et stratégiques du réseau. Cette partie de l'attaque peut être comparée aux attaques bien connues dans le domaine industriel : les *Advanced Persistent Threat* (APT). Dans l'attaque *Homing*, l'attaquant cherche à prendre la place du nœud stratégique par exemple en manipulant le processus d'élection qui permet de déterminer qui réalisera ce rôle. Il peut également essayer de le localiser géographiquement afin de pouvoir le désactiver physiquement. Le but de cette attaque est de bloquer le trafic et de permettre la mise en place d'attaques plus évoluées. L'attaquant d'abord externe souhaite gagner l'accès au réseau.

Les attaques sur la vie privée ont donc pour but d'inférer des informations sur le réseau, son comportement et ses nœuds afin de donner un avantage à l'attaquant. Il pourra ensuite utiliser les vulnérabilités identifiées pour mener des attaques de sécurité plus efficaces et puissantes. Il peut également obtenir des informations sur les systèmes de sécurité déployés, ces derniers étant statiques, il peut mettre en place une attaque pour les contrer. La protection de la vie privée est donc un critère important dans le déploiement des WSN.

2.5.3 Etat de l'art des contre mesures

La première contre mesure à mettre en place contre les écoutes passives est l'utilisation du chiffrement. Il permet d'assurer la confidentialité des données collectées en rendant l'acquisition du *payload* inintelligible pour un attaquant externe.

Néanmoins, de par le format de la trame et suivant la couche à laquelle la sécurité est déployée, ce *payload* chiffré peut englober plus ou moins d'en-têtes. Le chiffrement permet donc d'assurer la confidentialité de certaines de ces métadonnées.

Par exemple, le *payload* de la couche Réseau inclut toutes les informations des couches supérieures, métadonnées de ces couches comprises. Ainsi chiffrer au niveau Réseau permet de rendre inaccessible les

données des en-têtes des couches supérieures (Transport et Application). Néanmoins, les métadonnées des couches inférieures (Réseau, MAC et Physique) seront toujours disponibles.

Dans l'étude [22] menée par Hennebert et al., il est expliqué que le chiffrement peut être déployé à trois niveaux. Le choix de la couche où le mécanisme de sécurité va être déployé dépend de l'application mais également de l'interopérabilité recherchée. En effet, de par la nature contrainte des capteurs, il n'est pas possible de mettre en place un chiffrement à chaque couche et le choix de celle-ci doit se faire en accord avec les performances réseaux désirées et le cas d'utilisation prévu.

(a) Sécurité niveau MAC.

(b) Sécurité niveau Réseau.

(c) Sécurité niveau Application.

FIGURE 2.3 – Sécurité par couche.

- Chiffrement couche MAC (Figure 2.3(a)) : Lorsque l'on déploie la sécurité dans les WSN au niveau MAC, on dit alors que la sécurité est *hop-by-hop* c'est à dire que chaque nœud recevant le message doit connaitre la clé pour déchiffrer le message et pour ensuite le router si besoin. Ce mode peut être mis en œuvre avec n'importe quel réseau et protocole de communication pour les couches hautes. Assurer la sécurité à ce niveau offre un maximum de sécurité dans le WSN même lors d'écoutes passives. Dans ce mode de protection seul l'en-tête MAC ne peut pas être caché (en gris clair sur la Figure 2.3(a)) et apporte des informations de vie privée. Les autres en-têtes (en gris foncé) sont chiffrés.
- Chiffrement couche Réseau (Figure 2.3(b)) : L'un des challenges du déploiement à grande échelle des réseaux WSN concerne notamment la sécurité de bout-en-bout, c'est-à-dire, assurer la continuité de la sécurité dans la communication entre deux terminaux IP. Dans cette configuration, le même niveau de sécurité est utilisé de part et d'autre des réseaux. La passerelle a alors pour rôle de router les trames d'un monde à l'autre en aveugle. Ainsi, une compromission de la passerelle, ne met pas à mal tout le réseau. Dans ce mode, les adresses IP sont en clair afin de permettre le routage. Internet Protocol security (IPsec) est l'une des solutions envisagées pour la sécurité couche Réseau. Il est indépendant des couches Transport et Application mais n'est compatible qu'avec des réseaux IP. Une telle solution est appelée par le noyau de l'OS, ce qui le rend transparent pour un développeur d'applications et permet d'assurer une meilleure utilisation de la sécurité. Il permet également d'empêcher un grand nombre d'information d'être disponible mais laisse plus d'en-têtes visibles qu'une solution couche MAC, ce qui représente encore des fuites d'informations exploitables.
- Chiffrement couche Application (Figure 2.3(c)) : La sécurité Datagram Transport Layer Security (DTLS) mise au niveau Application permet de protéger les messages entre deux applications et fonctionne avec UDP. Elle complémente Transport Layer Security (TLS) qui elle fonctionne avec TCP. Elle permet d'assurer une sécurité *application-to-application*. En revanche, du fait que la sécurité soit mise au niveau Application, celle-ci est à la charge du développeur. Elle doit donc être pensée au moment du développement de celle-ci et appelée par celle-ci. Cela peut poser problème lorsque le développeur n'a aucune connaissance en sécurité. Enfin, dans cette configuration, les en-têtes des couches basses sont visibles représentant des fuites importantes car elles comprennent de nombreuses métadonnées.

Une dernière contre mesure consiste à mettre en place une solution de sécurité au niveau de la couche Physique. Saad, Zhou et al. dans [52] expliquent que ce sujet à fait l'objet de nombreuses recherches dernièrement et ils donnent un inventaire des différentes techniques permettant d'assurer la sécurité à cette couche.

Il est ainsi possible de déployer le chiffrement au niveau de la couche Physique. Dans ce cas, seul l'en-tête nécessaire à la réception, la démodulation et la synchronisation du signal RF est accessible.
Dans le cas de l'utilisation d'une clé de groupe, cette solution permet de lutter contre un attaquant externe et de ne laisser aucunes informations disponibles dans les en-têtes des couches supérieures. Néanmoins, un nœud légitime devra déchiffrer la trame avant de savoir si celle-ci lui est destinée ou non. Il va donc consommer de l'énergie inutilement. Avec le chiffrement couche MAC, les métadonnées permettent de filtrer la trame avant de commencer le travail de déchiffrement économisant l'énergie.
De même, si l'on souhaite contrer un attaquant interne, il est nécessaire d'utiliser des clés différentes pour chaque paire de communications. Or, comme aucun en-tête n'est disponible, un nœud devra faire une recherche exhaustive parmi les clés qu'il stocke afin de trouver la clé correspondant à l'émetteur et déchiffrer la trame. Ce travail devra être réalisé pour chaque trame reçue, qu'elle lui soit destinée ou non. Il va donc consommer de l'énergie à déchiffrer des trames. Avec un chiffrement couche MAC ce problème ne se pose pas car le nœud possède assez de métadonnées pour sélectionner la bonne clé. Les trames ne lui étant pas destinées seront également filtrées avant le déchiffrement.
Une solution consiste donc à ajouter un nouvel en-tête indiquant dans ses métadonnées l'émetteur et permettant ainsi d'identifier la clé à utiliser pour le déchiffrement mais également le récepteur afin d'éviter le déchiffrement inutile. Dans ce cas, l'utilisation du chiffrement couche Physique n'a plus d'intérêts car il faut ajouter un nouvel en-tête non prévu par les standards et apportant des informations de vie privée. Enfin, un attaquant peut, même lorsque que le chiffrement est déployé à la couche Physique, utiliser les spécificités physiques (RSSI, canal, qualité du lien...) pour identifier un nœud parmi d'autres.

Le chiffrement MAC apparait donc comme la meilleure solution contre l'écoute passive. Elle permet de masquer un maximum d'information accessible mais laisse malgré tout l'en-tête MAC en clair. La protection contre les écoutes passives est donc incomplète et compliquée à assurer. Ces informations peuvent encore être utiles pour mener une attaque par analyse de trafic.

En effet, même lorsque la trame est chiffrée, les activités des nœuds et du réseau peuvent être retrouvées grâce aux informations obtenues par analyse de trafic. Bernaille et al. dans [53] analysent le trafic d'applications web chiffrées avec Secure Sockets Layer (SSL). Ils démontrent ainsi une efficacité de 85% dans la reconnaissance obtenue grâce à l'utilisation des données statistiques des différents paquets échangés.

Il est donc nécessaire de déployer, en parallèle, des contre mesures adaptées à l'analyse de trafic.

Dans [54], Muntwyler, Lenders et al. proposent d'offusquer les communications au niveau de la couche Physique. Pour cela, ils utilisent une puissance d'émission calculée en sorte qu'un attaquant ne puisse pas détecter l'échange ou s'il y arrive, ne puisse pas le comprendre. Néanmoins, cette technique est très liée aux capacités de l'attaquant et au matériel qu'il utilise. De plus, cette contre mesure nécessite l'adaptation du matériel du commerce. En effet, la solution retenue utilise de nouveaux codes d'étalement de spectre pour réaliser la protection de la vie privée à la couche Physique. Il est donc nécessaire d'utiliser du matériel sur-mesure (*Software Defined Radio*).

Il est donc plus intéressant pour le moment de travailler sur des méthodes qui s'adaptent aux standards actuels en travaillant sur les couches au-dessus de la couche Physique. Ainsi, seules des modifications du code embarqué ou des protocoles utilisés sont nécessaires, ce qui facilite l'implémentation dans les systèmes actuels.

Afin de contrer l'utilisation des données de temps de transmission d'un message, la solution la plus efficace est de mettre en place une solution de type "mixeur". Le principe du mixage ou des réseaux de mélange a été introduit par Chaum en 1981 [55] afin d'assurer l'anonymat dans l'envoi de mails. Il permet de dépersonnaliser les données avant leur envoi ou le routage. Ce mode de routage utilise des serveurs intermédiaires appelés mixeurs routant l'information sans connaitre ni la source ni la destination finale de celle-ci. Elle a été prévue pour pallier aux attaques internes de type Man In The Middle (MITM) et à l'analyse de trafic.

Les mixeurs sont des blocs ou nœuds qui prennent plusieurs messages en entrée provenant de différentes sources. Ces messages sont ensuite modifiés avant routage par le nœud mixeur de manière à empêcher un observateur externe de lier les messages d'entrée avec les messages de sortie. Ils sont ensuite transmit soit à un autre nœud mixeur soit à la destination.

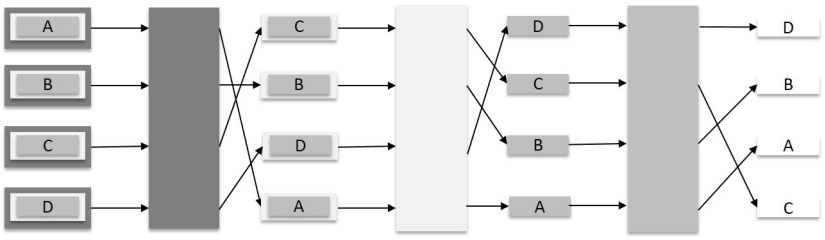


FIGURE 2.4 – Fonctionnement d'un réseau de mélange.

Les mixeurs vont changer l'apparence du paquet mais également modifier le débit de transmission. Diaz et al. dans l'article [56] font l'inventaire des différentes méthodes permettant de réaliser ces modifications.

Avant envoi, chaque message (A à D à gauche sur la Figure 2.4) est chiffré successivement avec la clé publique de chaque mixeur (ici 3 nœuds mixeurs) participant au routage jusqu'à l'adresse destination. Chaque niveau de gris correspond à une couche de chiffrement réalisée avec une clé différente. Le message est ensuite envoyé au premier nœud. Dès réception, le mixeur déchiffre sa couche de chiffrement à l'aide de sa clé privée afin de découvrir la destination du prochain saut pour le routage. Ainsi, les mixeurs n'ont connaissance que des sauts *one-hop* et l'apparence du message est différente entre chaque routage.

Le débit est également modifié afin d'empêcher l'analyse de trafic. On va alors ajouter du retard et réarranger les paquets. Cette modification est plus difficile à mettre en place et peut engendrer des retards dans des applications temps réel. Il est essentiel de clairement spécifier le pouvoir de l'attaquant avant de choisir quelle méthode sera appliquée. En effet, les auteurs de [56] donnent trois méthodes pour modifier le débit. L'une d'entre elle, exploitée par Kamat, Xu et al. dans [57], consiste à temporiser les messages dans des mémoires à chaque nœud mixeur et de les libérer selon une loi de probabilité. Ils expliquent qu'il est possible d'adapter le modèle utilisé pour le retard suivant la position dans l'arbre de routage évitant ainsi qu'un nœud proche du puits ait une mémoire saturée. Ainsi, ces nœuds pourront router un message préalablement stocké avant la fin normale du retard afin de libérer de l'espace pour un nouvel arrivant. D'après l'analyse faite dans [56], ce schéma peut malgré tout fournir des informations de vie privée dans des réseaux avec peu de nœuds. Il est approprié pour un réseau stable et à débit constant. Néanmoins, ce schéma revient souvent, sous des formes modifiées, dans les solutions contre l'analyse de trafic temporelle et est parfois combiné à des solutions pour contrer l'analyse du taux de paquets.

Pour contrer l'analyse du taux de paquets, la solution la plus étudiée consiste à introduire du faux trafic. Ce trafic va être généré par des nœuds qui n'ont pas d'informations à émettre et ne va pas être destiné à la station de base afin de tromper l'attaquant. Cette technique est également efficace pour détecter des attaques sur le réseau comme l'attaque n-1.

L'attaque n-1 est très utile pour un attaquant pour contourner les contre mesures de corrélation temporelle utilisant des nœuds mixeurs comme ceux de la solution précédente. En effet, lors d'une attaque n-1, un attaquant interne va être capable de retarder les messages qui vont au nœud mixeur. Il va alors choisir un message à tracer et va retarder les autres.
La solution pour lutter contre cette attaque consiste à injecter du faux trafic. Ainsi, un faux message peut être routé et renvoyé à sa source. Si la source reçoit moins de message qu'il en a envoyé, il peut penser à une attaque n-1. Néanmoins, cette solution contre l'attaque n-1 est compliquée dans le cas de réseaux avec pertes de paquets comme les WSN.

Lors du déploiement d'une solution contre l'analyse du taux de paquets, il est nécessaire de choisir la méthode de génération du faux trafic [56] la mieux adaptée à l'application déployée. En effet, l'envoi des faux messages peut être généré de façon indépendante ou non du trafic réel.

Dans l'article [58], Shbair, Bashandy et al. proposent de générer du faux trafic à partir d'échantillons de trafic réel afin de cacher l'inactivité d'un nœud. Grâce à cette méthode, l'attaquant n'arrive alors pas à différencier le vrai du faux trafic et donc à retrouver l'activité d'un nœud. Une première phase est nécessaire afin d'extraire des données sur la taille des paquets, le temps de transmission mais également les identifiants tels que les adresses IP ou les ports UDP.

C'est également une technique similaire qu'utilisent Mehta, Liu et al. dans [59] pour protéger la localisation de la source ayant captée l'évènement. Dans cet article, deux méthodes sont proposées.
La première, la plus optimale pour la protection de la localisation, consiste à ne pas envoyer l'information dès que celle-ci a été captée mais sur intervalles périodiques. Lorsque le *timer* est terminé, tous les nœuds émettent qu'ils aient ou non une donnée réelle. Pour cela, des faux paquets sont ajoutés. Cette technique est problématique pour les applications temps réel et nécessite une synchronisation.
La deuxième solution, quant à elle, nécessite un compromis entre protection de la vie privée et surcoût de communications. Dans celle-ci, des objets virtuels sont simulés. Pour cela, lors du déploiement du réseau, certains nœuds vont être dotés d'un *token*. Chacun des nœuds possédant le *token* vont alors émettre un signal imitant le signal de l'objet réel. Les voisins vont alors capter ce signal et le traiter comme réel. Le *token* est ensuite envoyé à un autre voisin pour qu'il effectue le même travail. Cette solution est intéressante pour les applications temps réel mais ne couvre pas la protection de la localisation de la station de base.

Shao, Yang et al. proposent dans l'article [60], que l'envoi ne soit pas périodique mais suive une loi statistique (loi exponentielle). L'envoi de faux paquets est programmé pour suivre cette loi et ne pas trop s'en écarter statistiquement. Dès qu'un vrai évènement est capté, la prochaine émission est recalculée de façon à respecter la loi statistique mais également à réduire la latence. Ainsi, un attaquant observant les communications ne peut extraire de celles-ci quel trafic est réel et lequel est simulé même avec la connaissance de la loi utilisée. De même que pour la solution précédente, cette contre mesure n'a pas d'impact sur la protection de la station de base.

Pensée en premier lieu pour améliorer la résilience des protocoles de routage face à un attaquant, la solution décrite dans [61] par Erdene-Ochir, Minier et al. peut également permettre la protection contre l'analyse de trafic. Dans cette solution, le protocole de routage Gradient Based (GBR) [62] a été modifié afin de permettre son fonctionnement même en présence d'un attaquant ayant corrompu entre 10 et 50% des nœuds du réseau. Dans GBR, les nœuds possèdent un gradient indiquant leurs distances vis-à-vis du nœud puits. Plus le gradient est élevé, plus le nœud est éloigné. Ce processus permet de connaitre le plus court chemin pour atteindre le puits. De par la nature du réseau, ce chemin reste identique. Un attaquant menant un attaque *selective forwarding* aura alors de forte chance de se trouver sur le chemin et donc de nuire plus facilement à la qualité de services. Pour pallier à ce problème, les auteurs ont modifiés GBR de façon à ce que le choix du prochain saut se fasse de manière aléatoire parmi l'ensemble des voisins les plus proches. Ce routage aléatoire est complété par une réplication des paquets. Les auteurs proposent de mettre de la redondance dans l'envoi en permettant aux nœuds routeurs et/ou au nœud source d'émettre le paquet plusieurs fois. Les différents paquets sont alors routés jusqu'au nœud puits via des chemins aléatoires différents. Ce routage aléatoire combiné à des paquets redondants permet de contrer l'analyse temporelle mais également l'analyse du taux de paquets. Dans le cas où les routeurs intermédiaires répliquent également les messages, cela permet de protéger la source de l'évènement. En revanche, cette solution ne protège pas la station de base voire accentue la détection de celle-ci.

Quatre contre mesures sont alors proposées par Deng, Han et al. dans [50] pour cacher la localisation de la station de base et de la source simultanément. Elles jouent sur le chemin utilisé pour rejoindre la station de base en ajoutant soit de l'aléatoire dans le routage, soit des faux chemins réalisés par des faux paquets. Dans cette dernière contre mesure, quand un nœud s'aperçoit que l'un de ses voisins transmet une donnée réelle, il émet un faux paquet dont la destination finale n'est pas la station de base mais se situe à k-sauts. Ce nombre de sauts doit être équivalent à celui nécessaire pour atteindre la station de base. Les voisins de ce même nœud vont à leur tour, lorsqu'ils vont capter le faux paquet, émettre un faux paquet mais à (k-1)-sauts. Dans la Figure 2.5, le nœud rouge souhaite envoyer une donnée réelle collectée au nœud puits vert qui se trouve à k=4 sauts. Il va alors l'émettre au premier saut. Quand le nœud A reçoit la trame, il la transmet au saut suivant. Le nœud violet est un voisin du nœud A. Lorsqu'il s'aperçoit que le nœud A va transmettre une donnée réelle, il crée un faux paquet qu'il destine à un faux nœud (le nœud orange) en respectant le même nombre de saut nécessaire pour atteindre la vraie destination (ici k = 2 sauts). Ainsi le véritable chemin sera noyé sous les autres. Une analyse temporelle sera alors inutile car il n'y aura aucun moyen de dissocier le vrai chemin des faux et donc de remonter à la véritable station de base, de même que pour l'analyse du débit. Néanmoins, l'introduction de paquets supplémentaires apporte un surcoût en émission et en traitement sur des nœuds contraints et dégrade fortement la bande passante. De plus, cette contre mesure n'empêche pas l'analyse des identifiants afin de cibler le vrai chemin.

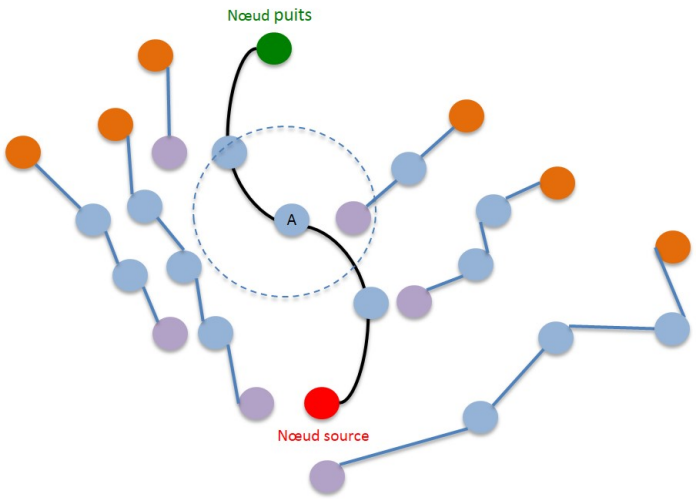


FIGURE 2.5 – Solution de vie privée à k-1 sauts.

Cette dernière faille due à l'analyse de trafic a été moins étudiée que les précédentes. Néanmoins, il existe quelques solutions. Les solutions proposées dans la littérature ont souvent du mal à protéger la vie privée de bout en bout s'autorisant, par exemple, à divulguer les identifiants lors du dernier lien avec la passerelle rendant le contrôle d'accès faisable.

Boukerche, El-Khatib et al. dans [63] proposent une solution de routage anonyme basée sur les réseaux de mélange ainsi que sur une autorité de confiance qui permet de ne sélectionner que les nœuds de confiance pour le routage. Leur solution appelée SDAR pour *Secure Distributed Anonymous Routing* utilise une autorité de certification de confiance qui permet de générer les clés privées et publiques du réseau. Contrairement aux réseaux de mélange, les techniques de modification de débit ne sont pas utilisées. Seule l'apparence est modifiée par le chiffrement. Un nœud source souhaitant communiquer avec un nœud destination va tout d'abord chercher à établir le chemin pour router sa trame. Pour cela, une technique réactive de détection de voisins est utilisée via l'envoi d'un message broadcasté. Une fois ce chemin établi, le nœud émetteur va chiffrer, de manière successive, le message de départ. Puis il va l'envoyer au premier saut. Chaque nœud intermédiaire va peler la couche de chiffrement lui correspondant faisant apparaitre l'adresse du prochain saut. Ce fonctionnement est appelé routage oignon. Contre un attaquant externe ayant une vision complète du réseau ou un attaquant global, cette solution ne protège pas l'adresse de la source lors du premier saut, ni l'adresse de la destination. De même, si un attaquant possède une vision de tout le réseau, les adresses *hop-by-hop* étant nécessaires pour router la trame, il pourra suivre le chemin et retrouver les adresses. Cette solution offre de l'anonymat uniquement pour un attaquant interne. Enfin, l'utilisation d'une clé privée induit des complexités de calcul importantes lorsque le nombre de messages broadcastés pour établir la route est important.

La solution présentée dans [64] par Nezhad, Miri et al. propose de remplacer les identifiants par des labels ou pseudonymes. Lors de l'initialisation, tous les nœuds du réseau y compris le nœud puits possèdent une clé partagée pré distribuée. Lors de la découverte de la topologie, le puits a une vision globale du réseau. Il va alors calculer la route pour atteindre chacun des capteurs. Chaque capteur ne possède qu'un père mais peut avoir plusieurs fils. Le puits va alors assigner des labels aux différents nœuds, lui compris. Ce pseudonyme va permettre de remplacer les adresses dans les communications. Ainsi un nœud va posséder deux labels. Un label pour les communications où il sera la source et un pour les communications où il sera destination. Le label sortant d'un nœud est le même que le label entrant de son nœud père. Par exemple, dans la Figure 2.6 le nœud rouge possède le label L6 pour ses communications entrantes et le label L1 pour les sortantes. Pour les voisins du puits, les labels sortants de ceux-ci sont différents de façon à ne pas apporter d'information à un attaquant. En effet, si tous les labels sortant des voisins du puits étaient identiques afin de l'adresser, comme pour le reste du réseau, alors un attaquant pourrait voir un grand nombre de trames avec le même label et par analyse du taux de paquets, en déduire la localisation de celui-ci. Il est également possible de

réutiliser des labels du moment où la distance entre deux nœuds ayant le même label est d'au moins 2 sauts. La distribution des labels se fait via un seul message par branche principale contenant tous les labels et grâce à un fonctionnement similaire au routage oignon permet à chaque nœud de récupérer leur label avant de router la trame. Ce schéma apporte un surcoût important lors de l'établissement du réseau et pour la distribution des pseudonymes. S'il est nécessaire de réattribuer des pseudonymes, cela entraine une nouvelle procédure assez lourde pour des nœuds contraints. De plus, le chiffrement avec les clés publiques de chaque routeur n'est pas réalisable avec tous les standards de l'IoT car augmente de façon considérable la taille de la trame et donc nécessite plus d'énergie pour l'envoi. Enfin, ce schéma est compliqué lorsqu'il existe des nœuds pouvant être mobiles.

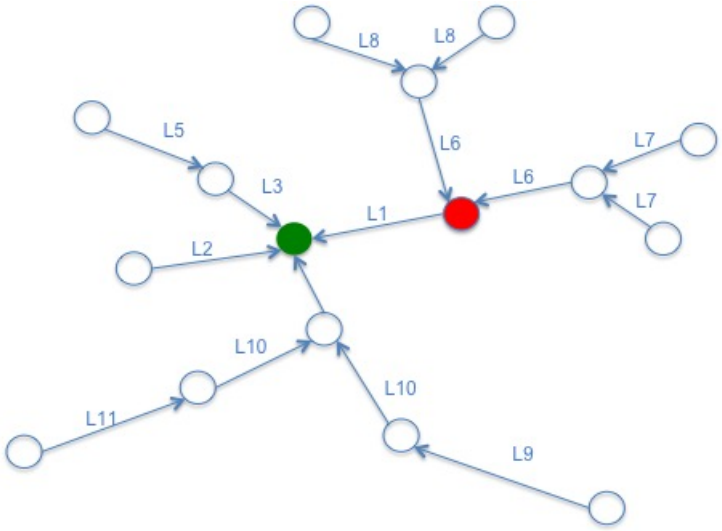


Figure 2.6 – Schéma d'utilisation des labels.

Une solution permettant de rendre anonyme tous les nœuds du réseau tout en permettant le contrôle d'accès est décrite par Huang dans [65]. Cette solution permet l'anonymat de l'utilisateur grâce à l'utilisation de pseudonymes. Un utilisateur peut posséder plusieurs pseudonymes. Pour cela, le schéma se base sur deux autorités : une autorité de confiance qui connait l'identité réelle mais pas le pseudonyme, et une autorité de certification qui connait le pseudonyme mais pas l'identité réelle. Cette solution permet l'anonymat au niveau de la couche Application du modèle OSI. L'autorité de confiance va fournir un ticket à l'utilisateur qui servira à générer le certificat du pseudonyme. L'autorité de certification, quant à elle, va fournir le certificat. Pour faire le lien entre pseudonyme et identité, il faut alors compromettre les deux autorités. Ce schéma est compliqué à mettre en place dans des réseaux vastes avec pertes de paquets. En effet, pour pouvoir communiquer et utiliser un pseudonyme, un nœud va devoir être à portée des deux autorités.

Dans [66], Debnath, Singaravelu et al. procèdent d'une manière différente. Ils utilisent les signatures de cercle (*ring signature*). Ce procédé cryptographique permet à un nœud de signer de façon anonyme un message. Cette signature est réalisée au nom d'un cercle et non au nom d'un seul individu. Le signataire choisi les autres membres du cercle. La signature est réalisée grâce aux clés publiques de ces membres.
Afin de cacher l'identité du capteur ayant relevé l'évènement, la signature du message est générée anonymement évitant ainsi de révéler l'identité du signataire. Pour cela, un ensemble de potentiel signataires est défini. Ce procédé permet d'authentifier le message sans pour autant révéler la source réelle.
Dans leur solution, le message est broadcasté et chaque nœud à portée va vérifier s'il appartient au cercle et si oui, il va accepter le message pour le router. Ainsi l'identité du nœud émetteur est préservée mais également la localisation de l'évènement. Cette technique est sensible à l'analyse de trafic par taux de paquets. En effet, un nœud peut appartenir à plusieurs cercles. Ainsi un nœud appartenant à plus d'un cercle doit être plus proche du puits ce qui permet de le localiser.

Il existe certaines solutions qui tentent de contrer les trois types d'analyse de trafic.

Dans l'article [67] Jiang, Wang et al. souhaitent cacher l'identité de l'émetteur mais également le temps de transmission et le RSSI. Afin de cacher l'identité, ceux-ci préconisent l'utilisation de pseudonymes dynamiques. Un nœud va alors demander à un point d'accès de lui affecter une adresse MAC. Cette demande ne se fait pas avec son adresse MAC réelle mais avec une adresse spécifique appelée "join address". Le point d'accès va alors choisir une adresse qui n'est pas encore assignée et répondre en incluant la "join address" et l'adresse MAC assignée. Le changement de pseudonyme intervient lorsque le nœud sait qu'il n'a plus d'information à communiquer. Ce changement dynamique évite l'analyse temporelle. Enfin, pour éviter la localisation, la solution propose de jouer sur le RSSI afin de ne permettre que la détection d'un seul point d'accès réduisant la possibilité de localisation par triangulation. Malheureusement bien que cette solution soit intéressante pour des réseaux où les points d'accès sont fixes et la portée importante, cette solution possède plusieurs inconvénients. En effet, comme pour la solution décrite dans [65], les capteurs n'ont pas forcément une station de base à portée. Leur demande doit donc être routée jusque celle-ci. De plus, il est possible que la demande soit perdue. Devoir compter sur une autorité afin d'obtenir un pseudonyme introduit de la latence et un surcoût de communications non négligeable dans des réseaux contraints. De plus, un attaquant peut déployer un réseau d'écoute passive important. La solution apportée pour cacher la localisation est donc inutile avec ce type d'attaquant.

Enfin, Luo, Ji et al. dans [49] proposent une solution permettant de contrer les trois formes précédentes d'analyse de trafic. Pour cela, il modifie le schéma de routage de façon à le rendre aléatoire. Il combine cette solution avec de l'injection de faux paquets et l'utilisation de l'anonymat. Leur solution souffre d'un manque de flexibilité dans un réseau mobile. De plus, il nécessite une phase de mise en place du routage importante qui, si le réseau est sujet à des pertes de paquets, peut prendre un temps important. Enfin, pour la régénération des pseudonymes utilisés, ils utilisent un nombre appelé "séquence" qui permet de vérifier la correspondance entre le pseudonyme reçu et celui calculé. Dès qu'une trame est reçue, le destinataire envoie un acquittement à la source et ajoute "un" à la valeur de la séquence. Dès que la source reçoit cet acquittement elle met à jour également la séquence. Comme il n'est gardé que la valeur courante du pseudonyme, si l'un des deux nœuds est désynchronisé, il ne pourra plus communiquer.

Cette dernière solution montre que les moyens de protection contre l'analyse temporelle et statistique semblent identifiés et malgré quelques problèmes de déploiement et de choix d'implémentation pourront à l'avenir protéger le WSN.

En revanche, les identifiants utilisés dans la dernière forme d'analyse de trafic sont plus compliqués à protéger. Ces identifiants sont utilisés pour le routage. De ce fait, ils sont transmis en clair pour autoriser le routage *hop-by-hop*, même lorsque le chiffrement est activé à la couche MAC. Il est alors facile pour un attaquant externe d'effectuer une écoute passive sur ces identifiants. Il faut donc mettre en place en parallèle une autre solution. Ces identifiants sont également utilisés dans certains mécanismes de sécurité comme les IDS ou pour le contrôle d'accès. Il faut donc que les solutions de protection ne perturbent pas ces mécanismes. Les contraintes particulières des WSN rendent les solutions de protection des identifiants difficiles à déployer. Le réseau est alors vulnérable à des attaques ciblées.

2.6 Conclusion

Les réseaux de capteurs permettent le déploiement et la mise en place de plus en plus d'applications dans des domaines variés. Toutefois, l'environnement atypique et les contraintes de ces derniers amènent de nouveaux problèmes de sécurité et de protection de la vie privée.

Les mécanismes classiques de sécurité tentent de s'adapter à ces nouveaux challenges en fournissant des solutions bas coûts pouvant être embarquées dans les nœuds de capteurs contraints. Néanmoins, des attaques sont toujours possibles et il est alors nécessaire de déployer des solutions dédiées afin de contrer ces nouvelles menaces sur la sécurité des WSN. De même, des solutions de prévention telles que les IDS sont pensées pour détecter ces attaques.

Malgré le déploiement de ces contre mesures, la nature sans fil des communications permet à un attaquant de collecter massivement et rapidement des informations de vie privée sur le réseau et ses participants. Il peut ensuite utiliser ces informations pour mener des attaques ciblées utilisant une vulnérabilité identifiée. Ces attaques sont facilitées par les métadonnées contenues dans les en-têtes nécessaires aux communications et au routage de la donnée à travers le réseau. L'une des solutions pour éviter la collecte de ces métadonnées est le chiffrement. Il permet suivant la couche où il est déployé de cacher les informations contenues dans les en-têtes des couches supérieures. Ainsi plus le chiffrement est déployé à bas niveau, plus il permet d'empêcher la collecte de métadonnées. Lorsque celui-ci est activé à la couche Physique, couche la plus basse, aucune métadonnée n'est disponible par écoute passive. Néanmoins, le déploiement apporte de la complexité pour le routage et le filtrage des trames et consomme beaucoup d'énergie. Il est alors nécessaire d'ajouter des en-têtes pour réduire cette complexité apportant ainsi de nouvelles métadonnées exploitables. Le meilleur compromis reste donc d'activer la sécurité au niveau de la couche MAC.

Lorsque le chiffrement est activé à la couche MAC, certaines métadonnées restent alors accessibles permettant la mise en place d'analyse de trafic. Des solutions spécifiques doivent donc être mises en place.

Assurer la protection de la vie privée dans les réseaux sans fil contraints est difficile et les solutions proposées dans l'état de l'art manquent souvent de maturité. C'est notamment le cas des solutions permettant de protéger les identifiants transmis dans les en-têtes d'une analyse de trafic. Ces identifiants sont indispensables pour l'adressage et le routage correcte des informations mais sont également utilisés dans certains mécanismes de sécurité comme le contrôle d'accès. La mise en place d'une solution de dissimulation de ces identifiants doit donc cohabiter avec ces différents mécanismes. Cette faille a été très peu étudiée et les solutions proposées ne sont pas optimales. Elles s'autorisent alors à divulguer les identifiants lors du dernier saut afin de permettre le contrôle d'accès au dépend de la protection de la vie privée.

J'ai donc choisi de travailler durant ma thèse sur la protection de la vie privée dans les WSN. Nous allons proposer une solution de protection des identifiants accessibles par simple écoute passive et utilisés pour mener des attaques actives. Cette solution prend en compte les différentes contraintes des WSN afin de ne pas nuire aux performances du réseau. Il est donc nécessaire d'identifier les contraintes mais également les identifiants à protéger. Pour cela, dans la partie suivante, nous allons décrire le contexte de l'étude.

Partie 3

La sécurité des protocoles radio basés sur IEEE 802.15.4

Dans cette partie, le standard IEEE 802.15.4 est présenté. Il est tout d'abord positionné vis-à-vis des protocoles radio grand public. Ses couches Physique et MAC sont décrites ainsi que les mécanismes de sécurité spécifiés dans la version publiée en 2006 du standard. Le fonctionnement et la sécurité du ZigBee et du 6LoWPAN, standards basés sur les couches IEEE 802.15.4, sont détaillés. Les briques logiciels et matériels utilisées dans cette thèse sont expliquées notamment le choix de Contiki OS, OS prometteur pour le déploiement de WSN.

3.1 Introduction

L'envie de développer de nouvelles applications, par exemple pour améliorer la domotique, les transports ou même les services municipaux, a entrainée de nouvelles contraintes. Des solutions "clé en main" étaient nécessaires afin de permettre le déploiement grand public de l'IoT.

Les capteurs peuvent être déployés dans un environnement bruité, être en mouvement mais également fonctionner avec des sources d'énergie très limitées.

L'idée n'est plus d'atteindre des performances en portée ou en débit comme avec les réseaux WLAN dont fait partie le WiFi mais plutôt de permettre à des objets de plus en plus petits et donc contraints de communiquer entre eux et cela sur des durées de vie importantes, et parfois d'être adressables depuis l'Internet.

Il a donc fallu trouver un protocole s'adaptant à ces nouveaux besoins.

La création et le déploiement faciles de ces réseaux sont des facteurs clés. Les données échangées ne sont plus de la voix, des images ou des fichiers qui nécessitent un haut débit mais des données courtes émises périodiquement ou sur évènement. La consommation d'énergie est également un critère pour le choix du standard des WSN.

L'intérêt pour les réseaux Wireless Personal Area Network (WPAN) n'a cessé de croitre que ce soit coté industriel ou coté recherche.

L'Institute of Electrical and Electronics Engineers (IEEE) a décidé de mettre en place un nouveau standard de communication répondant à des besoins de déploiement à large échelle et basse consommation.

En 2003, le standard IEEE 802.15.4 fait son apparition permettant le développement des Low Rate - Wireless Personnal Area Network (LR-WPAN). Il définit le standard en termes de couche Physique et de couche MAC. A la suite de cela, et notamment à cause de l'utilisation du standard par le grand public (pour la domotique, par exemple), l'IEEE 802.15.4 a évolué et une première révision a été publiée en 2006.

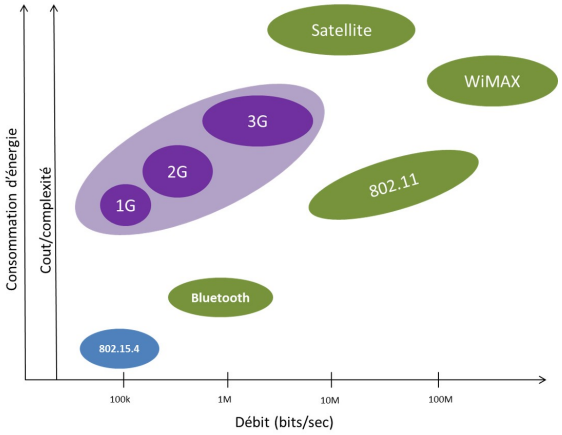


FIGURE 3.1 – Comparaison des consommations d'énergie des standards existants.

Caractéristiques	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.11
Autonomie sur piles	Années	Mois	Jour
Nombre de nœuds Dans le réseau	+65 000	7	256
Débit Moyen	250kb/s	1 Mb/s	10 à 1000 Mb/s
Portée (environ)	20 m	10 m	100 m

TABLE 3.1 – Comparaison des performances des standards.

Cette révision a permis une meilleure prise en charge de la sécurité et d'éviter certaines attaques du réseau. Afin de pallier aux besoins du monde industriel, un amendement a été publié en 2012 introduisant des modes de fonctionnement spécifiques.

Le standard IEEE 802.15.4 a été pensé, contrairement aux autres, pour fonctionner sur des réseaux WPAN. Comme le montre la Figure 3.1, l'IEEE 802.11 (i.e. le WiFi) possède un bon débit mais son coût en énergie ainsi que sa complexité font de lui un standard inadapté aux problèmes des WSN. Des objets contraints utilisant le WiFi auraient donc une grande portée mais ne fonctionneraient sur piles pas plus d'un jour. Kellogg, Talla et al. dans leur article [68] proposent une version du WiFi consommant environ 4 fois moins qu'un WiFi classique mais cette solution n'est pas encore standardisée. De plus, le WiFi reste identifié aux WLAN et ne permet pas le déploiement de réseaux meshés.

Le Bluetooth, quant à lui, possède une consommation d'énergie moins importante que le WiFi mais la durée de vie reste de l'ordre du mois ce qui est encore insuffisant pour ce type de réseau. De plus, la topologie piconet réduit les possibilités de déploiement. On dit que le Bluetooth est dédié WPAN courte portée mais haut débit (High-Rate WPAN). Enfin, le Bluetooth possède des temps de latence de l'ordre de 3s contre 15ms pour l'IEEE 802.15.4.

Le standard IEEE 802.15.4 présente un fort potentiel pour s'imposer en tant que standard des WSN. Cette thèse porte sur la confidentialité des données échangées dans les réseaux de capteurs IEEE 802.15.4 et notamment les identifiants transmis lors des communications.

3.2 Positionnement du standard IEEE 802.15.4 parmi les standards radio grand public

Les standards les plus courants dans l'IoT et notamment pour déployer des services utilisés dans les villes intelligentes sont présentés dans la Table 3.2. Ces standards ont des caractéristiques différentes et offrent des utilisations différentes de leurs données. Les fuites concernant leurs identifiants sont également différentes ce qui permet aux réseaux les déployant de protéger plus ou moins la vie privée de leurs nœuds et du réseau.

Technologie	Données	Référence	Expression	Couverture
GPS	Coordonnées géographiques	Absolue	Physique	Extérieure
WiFi	Id. du point d'accès + Force du signal ou coordonnées locales	Relative	Symbolique/ Physique	<100 m d'un Point d'accès
GSM	Id. + Force du signal ou coordonnées géographiques	Relative/ Absolue	Symbolique/ Physique	5 à 30 km
Bluetooth	Id.	Relative	Symbolique	5-10 m pour la classe 1 ; 20-30 m pour la classe 2
RFID	Id. du lecteur + position	Relative/ Absolue	Symbolique/ Physique	1m pour les RFID passifs ; 100 m pour les actifs
IEEE 802.15.4	Id.	Relative	Symbolique	10-20 m

Table 3.2 – Comparaison des 5 standards utilisés dans l'IoT.

- Le GPS : présent sur de nombreux appareils comme les smartphones mais également dans les véhicules, le GPS est une source intéressante de localisation dans un environnement extérieur. Les traces provenant d'un GPS ne sont pas assimilées à un appareil en particulier mais simplement à une localisation par rapport aux points cardinaux. Il est donc très difficile de remonter à l'identité de l'appareil à partir de ces coordonnées GPS. On peut donc penser que le GPS n'offre pas de métadonnées permettant l'analyse de trafic via les identifiants.

- Le GSM : également présent sur les téléphones portables, les traces récoltées grâce au GSM correspondent à l'identité de l'antenne relai, à la force du signal ou aux coordonnées géographiques de celle-ci. Ce standard de communication a pour avantage d'avoir une portée importante et est très présent dans une ville. Néanmoins, trop gourmand, il ne peut être déployé dans les nœuds contraints des WSN.

- La Radio Frequency IDentification (RFID) : fait partie d'une catégorie que l'on appelle capteurs flottants. Les tags RFID ont pour avantage de pouvoir être attachés ou incorporés sur de nombreux objets et permettent de les identifier, de les suivre et d'en connaître les caractéristiques à distance grâce à un tag émettant des ondes radio. La localisation par RFID se fait par le biais de l'identifiant du lecteur et de la position géographique de celui-ci. Après un état de l'art, il s'est avéré que de nombreuses recherches ont déjà été menées sur ce sujet comme celle menée par Sadikin et al. dans [69] et des solutions ont été trouvées pour combler les failles et protéger la vie privée des capteurs flottants. Il n'est donc pas abordé dans cette thèse.

- Le WiFi : standard de plus en plus démocratisé et que l'on retrouve dans de plus en plus d'objets de la vie quotidienne. Un appareil WiFi pour fonctionner, va chercher à se connecter à un point d'accès (*Access Point*, AP) qui va lui permettre de communiquer avec d'autres appareils WiFi et d'accéder au monde de l'Internet. De plus, il y a un grand nombre d'AP déployés dans une ville ce qui permet de relayer l'information. De nombreuses recherches ont été menées sur l'utilisation des métadonnées obtenues dans l'en-tête des trames WiFi. La majeure partie des fuites montre qu'il est possible de nuire à la vie privée de l'utilisateur. Néanmoins, peu de recherches ont été menées sur la protection de la vie privée de l'appareil WiFi. Une étude a été menée afin d'étudier ce que fournit l'en-tête d'une trame WiFi (cf. Annexe A).

- Le Bluetooth : utilisé dans les smartphones, le Bluetooth est également déployé dans les véhicules pour les kits mains libres par exemple, pour télécommander des avions ou des drones ou pour les montres connectées... Au fil des années, son utilisation auprès du grand public s'est vue réduite au profit de celle du WiFi. Néanmoins, il reste un standard sur lequel beaucoup de recherches sont encore menées et qui a pour ambition de fournir une nouvelle version basse énergie capable d'égaler sinon de remplacer le WiFi au sein de l'IoT. Un appareil Bluetooth fournit son identifiant de façon symbolique. Peu de recherches ont été faites sur ce standard et les identifiants contenus dans ses en-têtes. Néanmoins, les articles parus sur ce sujet comme l'étude réalisée par Takeda et al. dans [70] assurent qu'il est possible,

grâce à de l'écoute passive, d'obtenir l'adresse Bluetooth de l'appareil, c'est à dire un identifiant unique et donc par conséquence, il serait possible de suivre un appareil. Ce standard semble donc faillible et son analyse a été réalisée dans l'Annexe B .

- L'IEEE 802.15.4 : n'est pas un standard bien connu du grand public. Il commence à être utilisé pour les objets contraints et au sein des WSN comme par exemple dans les capteurs domotiques. Ce standard n'a pas été beaucoup étudié. Les nœuds IEEE 802.15.4 utilisent des identifiants exprimés symboliquement. Les réseaux communiquant grâce au standard IEEE 802.15.4 ont des topologies pouvant évoluer rapidement et les applications sont aussi nombreuses que différentes.

Parmi les standards détaillés ci-dessus et utilisés dans IoT, trois semblent porteur de failles concernant leurs en-têtes et ont fait l'objet d'une étude plus approfondie de leur norme. Dans la partie suivante, nous allons décrire le fonctionnement de l'IEEE 802.15.4. Nous allons décrire les différentes couches du standard mais également les protocoles et la sécurité utilisée. Puis nous détaillerons les solutions retenues pour les spécifications matérielles et logicielles.

3.3 Le standard IEEE 802.15.4 au service des WSN

3.3.1 Architecture des WPAN IEEE 802.15.4

Le standard IEEE 802.15.4 [71] a été créé dans le but de définir un nouveau standard de communication destiné aux réseaux LR-WPAN contraints en énergie disponible, en mémoire, en unité de calcul mais également en portée. Il est donc adapté aux WSN.

Les nœuds IEEE 802.15.4 peuvent être adressés soit par une adresse courte sur 16 bits, soit par une adresse de 64 bits appelée adresse MAC du nœud. L'adresse MAC (Media Access Control) correspond à un identifiant physique stocké par le nœud. Elle est liée au matériel et est unique.

Deux types de nœuds sont définis par la norme. Les Full Function Device (FFD) qui possèdent des caractéristiques moins contraintes. Ils peuvent ainsi assumer des rôles importants dans le réseau. Le deuxième type est appelé Reduced Function Device (RFD). Ceux-ci sont généralement plus contraints et n'ont pour but que de fournir des données à un FFD. Ils vont ainsi capturer des données (température, luminosité...) puis les envoyer à leur voisin FFD.

De la même manière, ces nœuds peuvent jouer différents rôles dans le réseau. Le premier et le plus important est le rôle du PAN coordinateur. Celui-ci est forcément assuré par un FFD. Il va initier le réseau, le gérer notamment lors d'ajouts ou de suppressions de nœuds. Dans certains cas, il peut également jouer le rôle de passerelle entre le monde des WSN et le monde de l'IP classique. Les nœuds RFD ne peuvent jouer que le rôle de nœud terminal (ou feuille) où, comme indiqué précédemment, ils vont récupérer les données via leurs capteurs et les envoyer à leur voisin. Enfin, le dernier rôle concerne les routeurs. Ces nœuds sont également des FFD qui ont pour but de router les paquets dans le réseau jusqu'à atteindre le PAN coordinateur. Ils vont également capter les informations de leurs environnements afin de les remonter.

Afin de permettre le déploiement de réseau vaste malgré une portée faible, le standard IEEE 802.15.4 supporte plusieurs topologies. L'une des topologies appelée *mesh* ou point-à-point, permet à tous les nœuds de communiquer entre eux. Pour cela, les routeurs vont permettre de faire circuler l'information. Cette topologie permet de ne plus être limité par la portée des nœuds et donc de déployer des réseaux plus denses mais nécessite de mettre en place un protocole de routage.

Le standard IEEE 802.15.4 ne définit pas le protocole de création de la topologie ni de routage ou de découverte des voisins. Cette tâche est laissée à la charge des couches supérieures. Seules les caractéristiques des deux couches les plus basses du modèle OSI ainsi que le mécanisme d'accès au média et la sécurité de la couche MAC sont décrites. Le management des clés cryptographiques utilisées pour la sécurité MAC est également délégué aux couches hautes.

Il est donc nécessaire de connaitre et d'identifier toutes les couches du modèles OSI fonctionnant avec le standard IEEE 802.15.4 et pas uniquement les couches basses.

3.3.2 Présentation de l'IEEE 802.15.4

Les caractéristiques nécessaires à la compréhension de la thèse sont données dans cette partie.

La plupart des OS actuels implémente la version 2006 [72] du standard IEEE 802.15.4 malgré la publication d'une révision en 2012. On peut noter que Kernel Linux 4.4 intègre depuis octobre 2015 la version 2012 du standard. Si aucune indication n'est donnée, la version étudiée est donc celle de 2006.

Pour de plus amples détails sur le standard, le lecteur est invité à consulter les standards ([71–73]).

3.3.2.1 La couche Physique

La couche Physique permet d'activer/désactiver la radio. Ainsi, un nœud pourra être en veille et économiser de l'énergie. Elle est responsable de la transmission et de l'émission sur le média. Elle met en place les techniques d'étalement de spectre ainsi que la modulation et la démodulation du signal. Elle va permettre de sélectionner le canal et faire de la détection d'énergie du signal sur ce canal. Elle va également permettre de calculer le *Link Quality Indicator* (LQI) après réception d'une trame afin d'évaluer la qualité du lien.

Suite aux dernières versions du standard, la couche Physique exploite maintenant 8 bandes ISM (*Industrial Scientific Medical*) dont les trois traditionnelles (de 2003) 868 MHz, 915 MHz et 2.4 GHz. 93 canaux se partagent ces bandes. Des bandes ont également été ajoutées afin de permettre l'utilisation de *l'Ultra Wide Band* (UWB) dans les techniques de localisation. La couche Physique permet également d'utiliser 7 modulations différentes.

Pour les bandes traditionnelles, 16 canaux se trouvent dans la bande 2.4 GHz, 10 dans la bande 915 MHz et 1 dans la bande 868 MHz. La bande 868-868.6 MHz est utilisée dans les pays européens, celle de 902-928 MHz est utilisée en Amérique du Nord et enfin la 2.4-2.483 GHz sert pour le monde entier.

Selon la bande utilisée, le débit peut aller de 20 à 250 kbit/s.

Dans la version de 2003, seul deux modes de fonctionnement existaient. En 2012, afin de pallier aux problèmes du monde industriel, 4 nouveaux modes sont venus s'ajouter. D'autres modes sont proposés dans la version 2012 et notamment une version *Low Energy* (LE). Néanmoins, par manque de spécifications des couches hautes à utiliser aux dessus de ces nouveaux modes, le déploiement n'est pas encore réalisé et réalisable.

Enfin la couche Physique gère l'accès au canal.

En effet, pour pouvoir communiquer, les nœuds doivent utiliser le protocole *Carrier Sense Multiple Access with Collision Avoidance* (CSMA-CA) afin d'éviter les collisions. Pour cela, un nœud va choisir un temps d'attente appelé *backoff period*. Dès que le temps est écoulé, le nœud va sonder le média. Il utilise le *Clear Channel Assessment* (CCA). Ce processus peut prendre différentes formes et permet de déterminer si le média est libre ou si un nœud voisin est en train d'émettre. Si celui-ci est occupé, le nœud repart sur une nouvelle période d'attente. Ce principe permet d'éviter un maximum de collisions même s'il n'est pas infaillible et qu'il est possible, si des demandes sont faites simultanément, de voir une collision.

3.3.2.2 La couche MAC

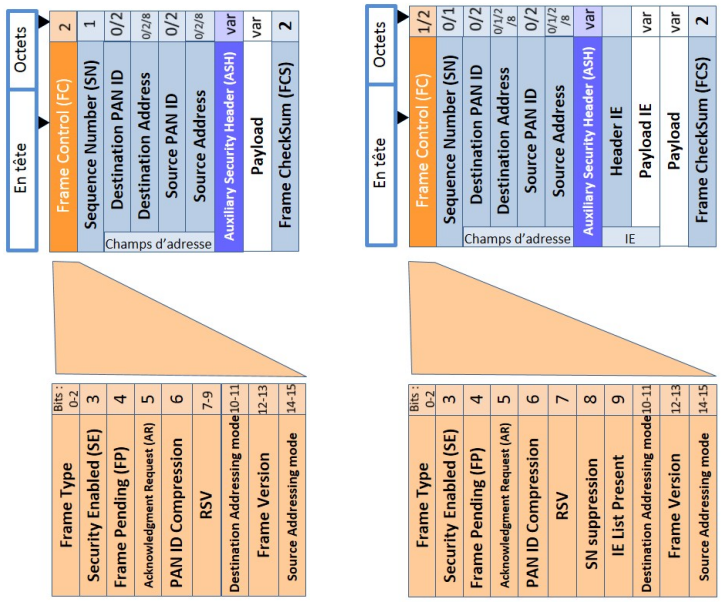
La couche MAC fournie une interface entre les couches hautes et la couche Physique. De ce fait, certains mécanismes décrit précédemment comme mécanismes de la couche Physique peuvent être considérés comme gérés par la couche MAC. C'est notamment le cas de la gestion de l'accès au canal ou encore le choix du *duty cycle* pour la veille.

Elle permet la validation des trames mais assure également les protocoles d'associations/désassociassions et de sécurité. Pour cela, la couche MAC (version 2012) définit six types de trames :

- Beacon et Enhanced Beacon
- Multipurpose
- LLDN

- Data
- Acknowledgment
- MAC command

Comme indiqué précédemment, seules les trames utiles pour la compréhension de la suite seront présentées. Les trames "LLDN", "Multipurpose" et "Enhanced Beacon" sont spécifiques au mode de fonctionnement introduit dans la révision de 2012. Toutes les trames MAC suivent le format général présenté dans la Figure 3.2. La taille maximale d'une trame IEEE 802.15.4 est de 127 octets.



(a) Format de la version 2006. (b) Format de la version 2012.

FIGURE 3.2 – Format général des trames MAC IEEE 802.15.4.

Dans la Figure 3.2, les deux versions des trames MAC (2006 et 2012) sont mises en juxtaposition.

Le champ SE du "Frame Control" donne des informations à propos de la sécurité. Si "SE" (*Security Enabled*) est à 1 cela indique l'utilisation de la sécurité couche MAC et la présence du champ ASH dans l'en-tête MAC, sinon celui-ci est absent. Le champ ASH sera détaillé dans la partie sécurité du standard IEEE 802.15.4. Ce champ permet de définir quelle sécurité est mise en place ainsi que les informations nécessaires pour générer les Initialization Vector (IV).

Les champs d'adresses présents dans l'en-tête permettent d'effectuer le routage *hop-by-hop*. Nous avons vu que les adresses représentaient une fuite importante d'information de vie privée pour les nœuds car elles pouvaient être utilisées pour mener des attaques ciblées ou faire de l'analyse de trafic.

Un champ "IE list present" est ajouté entre la version 2006 et 2012 dans FC afin d'indiquer la présence ou non des champs IE dans l'en-tête MAC. En effet, la version 2012 modifie quelque peu le format de l'en-tête MAC. Afin de permettre l'utilisation des nouveaux modes d'opérations définis dans la partie sur la couche Physique sans introduire de nouvelles trames ainsi que de permettre la mise à jour dans le futur du format des trames sans une refonte complète, le standard a ajouté un champ "Information Elements" (IE) de taille variable. Ce champ se compose de deux parties : l'une appartenant à l'en-tête MAC et l'autre à son *payload*. Ainsi, si besoin, les données contenues dans le *payload* peuvent être chiffrées à l'aide de la sécurité MAC.

Il est possible d'insérer un ou plusieurs champs IE. Il permet d'encapsuler l'information facilement. Il est également possible d'avoir des en-têtes sans *payload*. Cette amélioration de la norme est utilisée dans la thèse par la suite.

Pour pallier à une faille de sécurité existante jusqu'à la version 2012, la trame "Acknowledgment" a été modifiée. En effet, le premier format de la trame était pensé pour être le plus simple possible afin de ne pas occuper trop le média ni nécessiter trop d'énergie. Elle était alors composée uniquement des champs FC, SN et du FCS de la Figure 3.2. Comme aucune sécurité n'était possible avec cette trame, des attaques utilisaient cette faille pour forger de faux acquittements.

La nouvelle version de la trame d'acquittement permet d'insérer des champs d'adresses mais également de la sécurité. Néanmoins, ce processus de sécurité apporte des fuites d'identités. En effet, là où un attaquant passif n'avait pas d'information sur qui acquittait qui, avec le nouveau format, il peut suivre les échanges et reconstruire la topologie plus facilement.

La sécurité a évoluée entre la version 2003 et 2012. Certaines améliorations permettent, comme le chiffrement de la trame "Acknowledgment", de combler des failles. D'autres ont été pensées pour faciliter l'implémentation et la compréhension des mécanismes.

3.3.2.3 La sécurité

Le standard définit différents niveaux de sécurité. Il a pour objectif d'assurer :

- La confidentialité des données
- L'authenticité des données
- L'intégrité de la trame
- La protection contre les attaques par rejeu

Indépendamment du niveau choisi, la protection contre les attaques par rejeu est toujours activée.

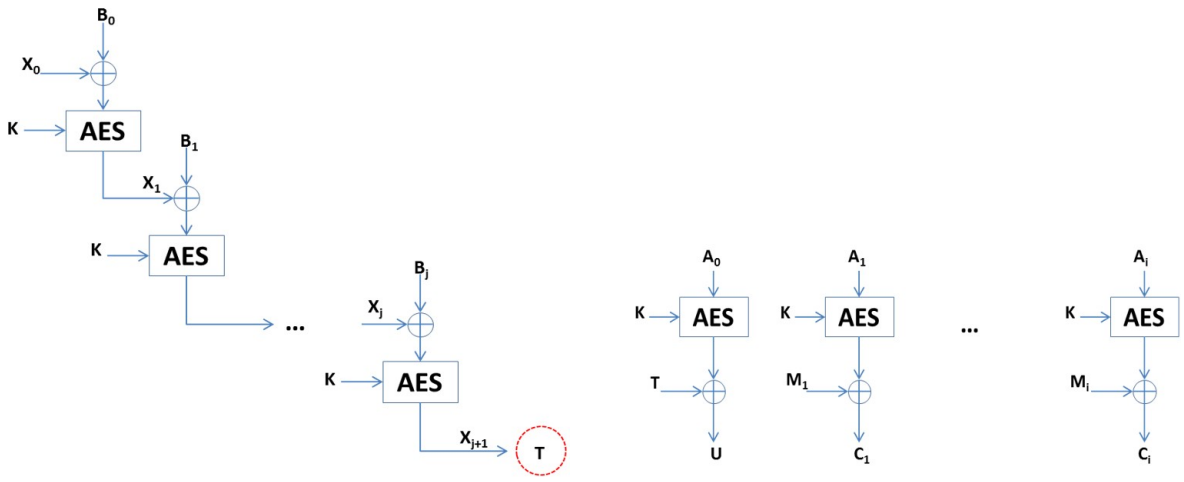
SL	Suite	Confidentialité	Authenticité
00	Non	Non	Non
01	MIC 32	Non	Oui (M=4)
02	MIC 64	Non	Oui (M=8)
03	MIC 128	Non	Oui (M=16)
04	ENC	Oui	Non
05	ENC-MIC 32	Oui	Oui (M=4)
06	ENC-MIC 64	Oui	Oui (M=8)
07	ENC-MIC 128	Oui	Oui (M=16)

Table 3.3 – Niveaux de sécurité.

Si un développeur choisi un niveau de sécurité égal à 0 alors aucune sécurité n'est déployée. Les niveaux de sécurité de 1 à 3 de la Table 3.3 permettent uniquement la mise en place de la vérification de l'intégrité de la trame alors que les niveaux de 5 à 7 permettent d'assurer la confidentialité et l'intégrité. Le niveau 4, quant à lui, ne permet que le chiffrement de la trame. Il est recommandé d'utiliser un niveau de 7.

Un seul et même algorithme est utilisé afin d'assurer tous les niveaux de sécurité. En effet, le standard utilise le chiffrement authentifié AES-CCM* qui combine le mode CTR de l'AES pour le chiffrement et CBC-MAC pour l'authentification. Dans une précédente version de la norme, l'AES-CCM était recommandé mais ne permettait pas de faire du chiffrement seul ou de l'authentification seule.

La version Advanced Encryption Standard-Counter with CBC-MAC (AES-CCM) possède des inconvénients qui limitent son utilisation. Rogaway et al. détaillent dans [74] certaines d'entre elles et donne une solution prenant le meilleur de AES-CCM et comblant les manques. Ce n'est pas cette solution qu'utilise le standard pour remplacer l'AES-CCM mais la version revue et standardisée. L'AES-CCM* permet également de combler des failles de sécurité existantes sur la version classique et que Fouque, Martinet et al. ont identifiées dans l'article [75]. L'une des vulnérabilité vient de la possibilité d'utiliser des tags d'authentifications de tailles différentes (entre 32 et 128 bits) suivant la nécessité. Or, ce fonctionnement le rend vulnérable à des



(a) Principe de l'authentification CBC-MAC. (b) Principe du chiffrement CTR.

FIGURE 3.3 – Mécanismes de sécurité du standard IEEE 802.15.4.

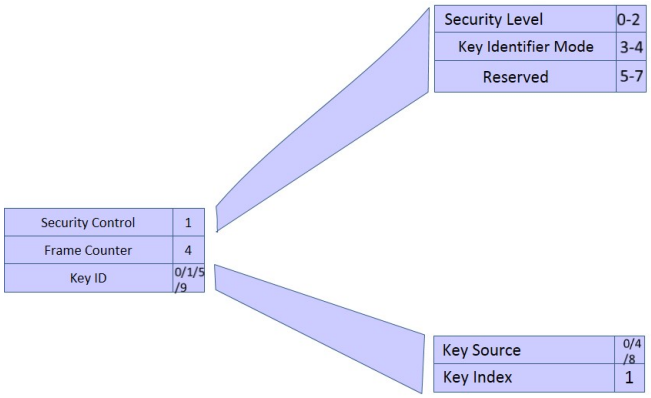


FIGURE 3.4 – Champs Auxilliary Security Header (ASH).

attaques spécifiques. En effet, du fait de la taille variable des tags, un noeud récepteur doit être configuré pour accepter des tags de toute taille. Un attaquant va pouvoir utiliser cette propriété pour forger des fausses trames et réussir à les authentifier.

Pour les niveaux de sécurité 5 à 7, le Message Integrity Check (MIC) est généré puis chiffré en même temps que le reste du *payload*. Lors de l'émission de la trame on authentifie donc d'abord celle-ci en ajoutant le MIC puis on assure la confidentialité. Lors de la réception, il faut d'abord déchiffrer la trame avant de s'assurer de son intégrité. Les schémas de principe de l'authentification et du chiffrement sont donnés dans la Figure 3.3.

Afin de savoir avec quel niveau de sécurité la trame a été traitée et comment les IV (notés B_i et A_i dans la Figure 3.3) ont été générés, le standard ajoute dans l'en-tête MAC le champ "Auxiliary Security Header" (ASH). Ce champ, décrit dans la Figure 3.4, est présent uniquement si le bit "Security Enabled (SE)" du FC est à 1.

Le champ "Security Control" de 1 octet permet d'indiquer comment et quel type de sécurité a été mis en place afin d'aider le récepteur à déchiffrer et/ou authentifier la trame.

Il comporte un premier champ "Security Level" permettant d'indiquer le niveau de sécurité établi parmi les 7 de la Table 3.3. Ainsi le récepteur sait s'il doit déchiffrer et/ou vérifier l'intégrité avant de traiter la trame reçue.

Le champ "Frame counter" permet de fournir une protection contre les attaques par rejeu. Enfin, le champ "Key ID" permet de spécifier les informations sur la clé utilisée.

Comme le montre la Figure 3.3, pour fonctionner, l'algorithme AES-CCM* a besoin de plusieurs éléments :

- Les données à chiffrer ou authentifier
- Les IV B_i et A_i
- La clé K

Les données à chiffrer et/ou authentifier vont dépendre du type de trame MAC émise.

Dans le standard IEEE 802.15.4, le parti prit est de ne pas transmettre l'IV dans la trame pour ne pas augmenter la taille de l'en-tête MAC et donc réduire le *payload*. Il faut alors que le récepteur de la trame puisse retrouver avec les données reçues comment celui-ci a été formé.

Les IV A_i utilisés pour le chiffrement suivent donc le format de la Figure 3.5.

FIGURE 3.5 – Format des IV pour le chiffrement.

L'élément le plus important de l'IV est le *nonce*. Il est lié à l'adresse étendue de l'émetteur de la trame, transmise en clair dans l'en-tête MAC. Nous verrons par la suite que l'utilisation de l'adresse MAC comme partie de l'IV complique le déploiement de solution de protection de l'identité des nœuds combinée au chiffrement.

Dans le standard la clé est définie explicitement ou implicitement et stockée. Une seule clé est utilisée. La gestion et le management des clés sont laissés à la charge des couches hautes. C'est pourquoi il est important de définir maintenant qu'elles sont ces couches hautes.

3.3.2.4 Les couches hautes du modèle OSI

Le standard IEEE 802.15.4 ne définit que les couches basses et laisse aux couches hautes la responsabilité de gérer nombres de protocoles important notamment ceux touchant à la sécurité.

Bien avant la publication de la version 2012 dédiée monde industriel, l'IEEE 802.15.4 avait déjà été choisi comme fondation de nombreux standards pour WSN déployés dans des environnements industriels. Le ZigBee ainsi que sa version ZigBeePro font partis de ces standards ayant optés pour l'IEEE 802.15.4 pour ses couches basses. Ils permettent de déployer des réseaux utiles pour des applications dans la domotique ou le *smart energy*. D'autres encore comme le WirelessHART ou l'ISA100.11a offrent des possibilités pour les automates industriels.

Malgré une efficacité prouvée dans le monde industriel, ces standards ne permettent pas d'établir simplement des communications IP entre capteurs. Or ce besoin grandit dû aux nouvelles applications de l'IoT. En effet, avec les communications IP, il est possible de fournir une adresse IP au nœud et d'utiliser celle-ci pour les requêtes. Ainsi, un utilisateur connaissant le déploiement de son réseau et les adresses peut, par exemple, consulter la température directement de son capteur déployé grâce à une requête lui étant adressée. C'est pourquoi le standard IPv6 Low power Wireless Personal Area Network (6LoWPAN) a été créé.

Dans cette thèse, les standards ZigBee mais également 6LoWPAN seront étudiés en terme de fuites d'identifiants. Il est donc nécessaire de connaitre le fonctionnement des deux ainsi que les protocoles utilisés.

3.3.3 Le ZigBee

Le standard ZigBee a été proposé par un consortium d'entreprises appelé "Alliance ZigBee". C'est un standard dit propriétaire qui permet le déploiement de WSN routé et meshé. De nombreux produits sont déjà commercialisés. Le modèle OSI du standard ZigBee décrit dans la Figure 3.6 consiste en plusieurs couches.

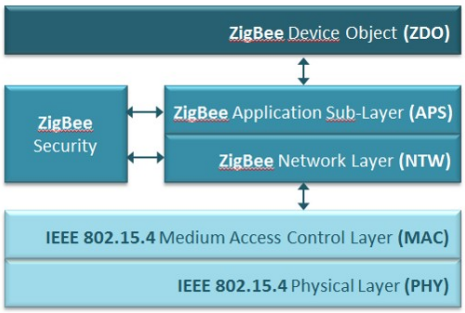


FIGURE 3.6 – Modèle OSI du standard ZigBee.

La première couche est la couche Réseau ("ZigBee NeTWork Layer, NTW"). Elle permet de définir les protocoles de routage utiles aux échanges multi sauts mais également de création des topologies ainsi que les protocoles d'adressage. Au-dessus de celle-ci, la couche Application ("ZigBee APplication Sub-layer, APS") offre des supports afin de définir les profils, *cluster* et *endpoints*. Les profils sont en fait des règles et réglementations qui permettent de créer une couche Application selon l'application voulue mais interopérable pour des appareils séparés. Il existe par exemple un profil pour la domotique ou encore la santé. Enfin, une couche "ZigBee Device Object (ZDO)" est définie. Elle fournit les services de découvertes et de gestion des objets ZigBee.

Les couches ZigBee offrent tous les protocoles nécessaires au bon fonctionnement du réseau. Ainsi, un développeur pourra se concentrer sur l'application. Dans ZigBee, les nœuds peuvent jouer 5 rôles différents. Le premier appelé ZC pour ZigBee Coordinateur a un rôle similaire au PAN coordinateur du standard IEEE 802.15.4. Il s'agit forcément d'un nœud FFD. Il existe au moins un ZC dans chaque réseau. Il permet de sélectionner le PAN ID ainsi que le canal de communication. Le deuxième rôle est celui du ZigBee Routeur (ZR). Enfin, le ZigBee End Device (ZED) dont le rôle est similaire aux nœuds terminaux définis précédemment. Après avoir rejoins le réseau par un ZR ou directement par le ZC, le ZED peut transmettre ou recevoir des données. Contrairement au ZED, le ZC et les ZR ne peuvent se mettre en veille et peuvent stocker les paquets entrant destinés à des ZED en veille. A ces trois rôles déjà définis par le standard IEEE 802.15.4, s'ajoutent deux autres rôles : la ZigBee Gateway (ZG) qui assure l'interopérabilité avec les autres protocoles de communications, elle permet notamment d'envoyer les données sur Internet et le ZigBee Trust Center (ZTC) qui est responsable de l'authentification des nœuds joignant le réseau. Il gère également la distribution des clés. Ces deux rôles peuvent être assumés par un ZC.

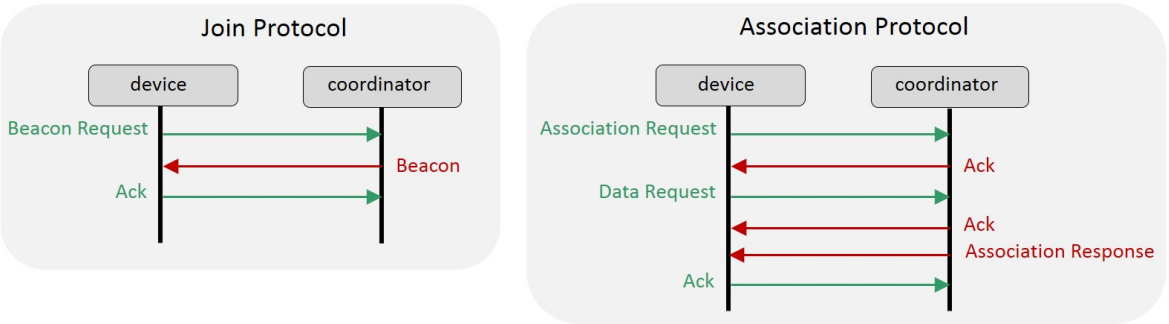


FIGURE 3.7 – Protocole du standard ZigBee.

Lorsqu'un nouveau nœud souhaite rejoindre un réseau ZigBee, il doit effectuer une phase dite de *join*. La première étape consiste à scanner le PAN grâce à l'envoi d'une trame MAC IEEE 802.15.4 de type "Beacon Request". Tous les ZR et le ZC appartenant déjà au réseau, s'ils sont proches, vont répondre par l'envoi d'une trame MAC IEEE 802.15.4 "Beacon". Cette trame contient les informations nécessaires pour rejoindre le réseau comme le PAN ID. Le nouvel arrivant peut acquitter la bonne réception de la trame comme le montre la Figure 3.7. Cette phase terminée, le nœud va pouvoir s'associer.

Le nouveau nœud va devoir choisir un parent parmi les émetteurs. Pour cela, il va envoyer une demande "association request". Le nœud parent va alors attribuer au nœud une adresse suite à la demande "data request" et confirmer son rôle de parent via l'envoi d'une trame "association response". Le nœud pourra ensuite communiquer.

Afin de communiquer dans le réseau, un nœud doit obtenir du coordinateur une adresse. Deux techniques d'adressage sont utilisées : aléatoire, le coordinateur possède une base d'adresses qu'il attribut jusqu'à épuisement ou hiérarchiquement. Dans ce cas-là, chaque routeur se voit attribué une partie de la base d'adresses qu'il peut ensuite attribuer à ses enfants. Pour cette solution, le coordinateur définit un nombre maximal d'enfant par nœud ainsi que la profondeur maximale du réseau. Bien qu'assurant l'unicité des adresses, cette technique peut engendrer des pertes d'adresses si un routeur ne possède aucun nœud à qui assigner la plage d'adresses qu'il possède.

La manière dont les adresses sont assignées induit un changement dans le protocole de routage. ZigBee offre donc deux protocoles de routage. Le premier, dans le cas où les adresses sont assignées aléatoirement, utilise une version allégée du protocole AODV. Dans cette solution un nœud souhaitant communiquer va d'abord regarder sa table de routage. Si l'adresse destination ne figure pas dans celle-ci, le nœud va alors faire une demande de route à l'aide d'une trame ZigBee NTW "Route Request" à ses voisins. Cette requête sera relayée jusqu'à atteindre la destination qui répondra alors avec une trame NTW "Route Reply". Dans le second schéma d'adressage, la topologie étant forcément de type arbre, les nœuds routeurs gardent dans leur table l'adresse de leurs parents et de leurs enfants. Les ZED ne gardent que l'adresse de leur parent. Dès qu'un nœud veut communiquer, il regarde dans sa table de routage si l'adresse lui est connue. Si ce n'est pas le cas il transmet à son parent qui fait de même jusqu'à atteindre la destination. Une trame "Link status" est émise périodiquement par les ZR et le ZC afin de maintenir les tables de routage à jour.

La Figure 3.6 montre que des mécanismes de sécurité peuvent être déployés à n'importe quelle couche du modèle OSI. Le standard ZigBee supporte plusieurs caractéristiques de sécurité. Il possède une implémentation logicielle d'un AES afin d'effectuer le chiffrement des données. Deux clés de sécurité sont également préconfigurées ou peuvent être obtenues durant la phase d'association grâce au ZTC.

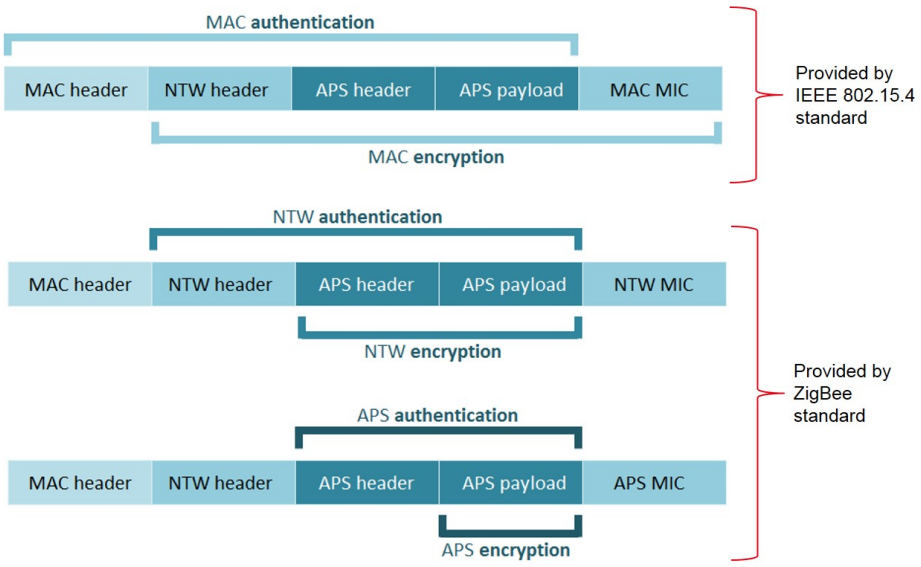


FIGURE 3.8 – Chiffrement et authentification grâce aux mécanismes définit dans ZigBee.

Trois types de clés sont introduits dans le standard :

- La clé dite "master". Préinstallée lors de la fabrication. Elle est utilisée pour générer les clés liens. Si celle-ci n'est pas fournie, l'adresse MAC est utilisée en remplacement.
- La clé lien permet de sécuriser les communications *peer-to-peer* au niveau de la couche Application (*payload* APS).
- La clé réseau est partagée par tous les nœuds du réseau. Elle permet de chiffrer la couche APS (en tête + *payload*) comme le montre le schéma du milieu de la Figure 3.8. Cette sécurité est utilisée également pour chiffrer les trames de routage dans le cas d'émission de "route request/route reply" ou encore les commandes ZDO. Néanmoins, elle est inutile pour les trames MAC de type "Beacon".

Si la sécurité est activée, tous les paquets seront chiffrés avec la clé réseau. La sécurité niveau APS est une sécurité *end-to-end*.

C'est le ZC qui sélectionne la clé réseau. Elle peut être préconfigurée ou sélectionnée aléatoirement. Le ZC joue généralement aussi le rôle de ZTC. Dans ce cas, il choisit la clé master. Un nœud souhaitant rejoindre le réseau doit obtenir la clé réseau pour communiquer. Si celui-ci possède une clé master préconfigurée, la clé réseau est alors envoyée chiffrée avec la clé lien qui dérive de la clé master. Sinon la clé est envoyée en clair.

Dans les réseaux de type WSN, un critère intéressant est la mobilité. En effet, il est essentiel de permettre à un nœud de pouvoir passer d'un réseau à un autre tout en maintenant les communications. Une solution serait d'utiliser le protocole *Network Address Translation* (NAT) qui permet de faire correspondre des adresses. Néanmoins, ce processus est coûteux. Il est donc essentiel de permettre aux nœuds de joindre plusieurs réseaux sans utilisation de NAT. De plus, dans l'IoT différents standards cohabitent et forment la chaine de communication allant du capteur à la base de données. Il est donc essentiel de permettre l'intégration de ces autres standards comme l'Ethernet sans trop de modifications.

C'est pourquoi l'utilisation de l'IP et plus particulièrement des adresses IPv6 est devenu obligatoire. ZigBee possède une couche Réseau propriétaire qui n'est pas adaptée à l'utilisation de l'IP. De plus, du fait que les couches ne proviennent pas de standards, l'implémentation de la *gateway* est complexifiée. L'interopérabilité doit alors avoir lieu au niveau du ZG. Ce ZG est complexe à réaliser car celui-ci doit, en plus de transformer des adresses IP en adresses ZigBee, adapter les commandes provenant des couches classiques de l'IP (UDP/TCP...) en commandes compréhensibles par un nœud ZigBee.

L'IETF a publié un nouveau standard, appelé 6LoWPAN, permettant l'utilisation de l'IP. ZigBee a profité de cette nouvelle norme pour créer un nouveau modèle OSI appelé ZigBee IP (ZIP) basé sur les différentes couches normalisées par l'IETF et l'IEEE. Ainsi, avec ce nouveau modèle, ZigBee permet le déploiement de protocoles comme UDP ou même TCP. Il déploie notamment 6LoWPAN et son protocole de routage.

3.3.4 6LoWPAN

Afin d'assurer les communications inter-réseau, un protocole permettant de supporter les communications Internet sur les WSN a dû être développé. En effet, le protocole IP étant utilisé et largement déployé depuis longtemps pour l'Internet, de nombreuses infrastructures sont disponibles et les outils ainsi que la sécurité déployée ont été éprouvés. Il est donc intéressant de pouvoir réutiliser ces outils ainsi que les adresses IPv6 dans l'IoT. Ainsi, s'il est donné à chaque nœud la possibilité de posséder une adresse IPv6, il sera alors plus facile de communiquer directement avec le nœud et ce indépendamment du réseau dans lequel on se trouve. Or, les protocoles classiquement utilisés sur l'IP souvent lourds et énergivores ne pouvaient être réemployés tel quel dans des réseaux avec de fortes contraintes. Comme l'expliquent Hossen, Kabir et al. dans l'article [76], la mise en place de l'IP dans le monde du contraint n'est pas trivial. Plusieurs challenges existent et c'est pour cela qu'il est obligatoire de définir un nouveau standard. Les groupes de standardisations IEEE et IETF ont donc publiés différents protocoles rendant l'IoT réalisable et permettant ainsi l'interopérabilité entre le monde de l'IP classique et le monde du LR-WPAN. Pour cela, la pile OSI de la Figure 3.9 a été développée.

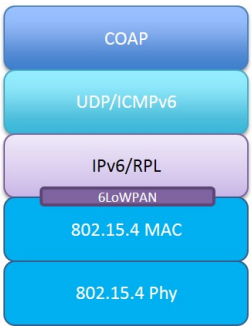


FIGURE 3.9 – Modèle OSI du standard 6LoWPAN.

Il a fallu alors résoudre les différents problèmes qui se présentaient. Du fait des applications espérées mais également de la nature des nœuds et des réseaux, le standard pour les couches basses s'est tout naturellement porté vers l'IEEE 802.15.4.

Le standard IP nécessite un Maximum Transmission Unit (MTU) de minimum 1280 octets. Dans les spécifications du standard IPv6, il est expliqué que, dans le cas où la couche lien n'accepte pas des paquets de cette taille, une couche intermédiaire doit être ajoutée pour permettre la fragmentation.

La taille d'une trame IEEE 802.15.4 est de 127 octets maximum. 6LoWPAN [77] a donc été introduit. Il permet de fragmenter les paquets si ceux-ci sont trop importants pour la couche MAC IEEE 802.15.4 mais également de les compresser. En effet, les en-têtes successifs peuvent contenir des informations redondantes ou pouvant être déduites. De plus, les adresses IPv6 étant sur 128 bits, elles tiennent une place importante dans les en-têtes réduisant la place allouée pour la donnée. 6LoWPAN permet via un en-tête ajouté de spécifier quels champs peuvent être déduits ou ont été supprimés et donc de réduire l'impact des en-têtes sur la taille de la trame.

IPv6 est un protocole internet spécifié par l'IETF en 1998 dans la RFC 2460 [78]. Il a pour but de succéder au protocole IPv4. En effet, les adresses IPv4, définies sur 32 bits, permettent d'allouer un peu plus de 4 milliards d'adresses. Or, le nombre d'objets connectés tels que les smartphones et tablettes, nécessitant une adresse IP, ne cesse d'augmenter. L'espace d'adressage disponible grâce à l'IPv4 est alors nettement insuffisant. IPv6 dispose de 128 bits d'adresse au lieu des 32 bits de l'IPv4, permettant d'augmenter considérablement l'espace d'adressage disponible.
L'adresse IPv6 se décompose en deux parties :
le préfixe sur 64 bits est une partie commune à toutes les adresses d'un réseau. Il peut prendre des valeurs spécifiques définies dans la RFC 2373 [79]. C'est la partie visible par un utilisateur externe.
La deuxième partie sur 64 bits s'appelle Interface IDentifier (IID). Elle est spécifique à l'objet ou au sous réseau. Différentes méthodes existent dans IPv6 afin d'allouer cette partie.

Ce découpage permet d'éviter l'allocation d'ensemble d'adresses par région comme cela était le cas avec IPv4 et une plus grande flexibilité dans l'adressage. Ces spécifications font de lui un protocole pratique pour des réseaux dynamiques, de structures complexes, nécessitant un routage tel que les réseaux de capteurs 6LoWPAN.

Dans un WSN 6LoWPAN deux types d'adresse sont utilisés :

- Une adresse dite *Link Local address* formée avec le préfixe FE80. Elle permet de communiquer au sein d'un même réseau.
- Une adresse globale. C'est une adresse unique qui peut être adressée à partir d'un autre réseau. Le préfixe doit prendre une valeur qui n'est pas une des valeurs réservées par IPv6 pour des usages particuliers.

Ces adresses permettent aux données enregistrées par les capteurs d'être communiquées soit à un utilisateur soit à une base de données en suivant un schéma de routage qui est mis à jour périodiquement. Ces adresses sont allouées lors de l'établissement du réseau.

Avec 6LoWPAN, le monde de l'IP classique est maintenant accessible plus facilement. Néanmoins, 6LoWPAN n'assure pas certaines fonctionnalités nécessaire à l'établissement et au maintien d'un réseau comme la configuration de ce dernier ou encore le routage des données.

C'est pourquoi l'IETF a publié la RFC 6550 [80] introduisant Routing Protocol for Low-Power and Lossy Networks (RPL), un protocole de routage *route over*. Le routage *route over* permet de router les trames au niveau de la couche Réseau (couche 3 du modèle OSI) comme cela est le cas dans les réseaux IP classiques. RPL permet ainsi à chaque nœud de se comporter comme un routeur IP et de réutiliser les capacités existantes dans l'IP classique notamment pour la configuration du réseau par l'utilisation des trames Internet Control Message Protocol version 6 (ICMPv6) [81]. Il supporte une grande variété de couches MAC et permet donc d'assurer l'interopérabilité entre deux réseaux dont les couches basses sont différentes. RPL va permettre au réseau de s'établir sous la forme d'un Destination Oriented Directed Acyclic Graphs (DODAG). RPL permet de créer et maintenir les tables de routage grâce aux trames ICMPv6. Il permet ainsi d'obtenir des communications *point-to-multipoint*, *multipoint-to-point* ou *point-to-point*. Néanmoins, RPL est optimisé pour un usage *multipoint-to-point*, cas d'usage fréquent dans les réseaux de capteurs où des nœuds vont collecter des données comme la température d'une pièce et vont remonter ces données vers un nœud puits ou un récepteur.

Dans les réseaux contraints, les pertes de paquets sont autorisées. Afin de pallier à ce problème, la couche Transport n'est pas TCP mais UDP facilitant la transmission des données. UDP ne nécessite pas de poigné de main pour établir une connexion mais n'assure pas de la bonne réception ni de l'ordre d'arrivée.

Enfin, les systèmes actuels de découverte de service tel que SOAP sont trop lourds pour être utilisés dans les WSN. Les WSN vont avoir une place importante dans les architecture RESTful. Ils vont interagir avec le web ou le cloud. L'IETF a donc défini Constrained Application Protocol (CoAP) qui transpose les demandes HTTP. CoAP utilise deux types de messages : "request" et "response". Il fonctionne au-dessus d'UDP.

L'un des grands avantages de la réutilisation de l'IP et des adresses globales IPv6 sur des nœuds contraints est qu'il est possible de mettre en place de la sécurité *end-to-end*.

Mais assurer la sécurité dans les WSN est encore plus critique que dans les autres environnements. En effet, les données émises ne sont pas de simples données mais peuvent être utilisées pour commander l'ouverture d'une porte ou éteindre des alarmes. Il est essentiel d'assurer la confidentialité et l'intégrité. Chaque couche peut implémenter les mécanismes nécessaires à la sécurité. Hennebert et al. dans l'article [22] décrivent les différents mécanismes disponibles.

Au niveau de la couche Réseau, IPsec peut être intégré. Ce protocole existe dans le monde IP classique. Il permet la mise en place d'une sécurité *end-to-end*. IPsec offre différents modes de sécurité. Le mode Encapsulating Security Payloads (ESP) permet de chiffrer les données avec ou sans authentification. Le deuxième mode Authentication Header (AH) fournit l'intégrité et l'authenticité. Afin de fournir les algorithmes et le matériel nécessaires pour effectuer les deux modes précédents, les nœuds utilisent le protocole Security Association (SA). Quel que soit le mode utilisé, IPsec peut être implémenté en mode tunnel ou en mode transport. Dans le mode transport, seul le payload IP est chiffré et/ou authentifié. L'en-tête reste identique et en clair. En mode tunnel, l'en-tête IP est chiffré/authentifié avec son payload et un nouvel en-tête est ajouté. Ce fonctionnement est similaire à l'utilisation d'un Virtual Private Network (VPN). IPsec assure la fraicheur des paquets mais ne fournit pas de support d'acquittement.

En l'absence d'IPsec, la sécurité à la couche Transport peut être assurée grâce à DTLS. Ce mécanisme est basé sur TLS, mécanisme présent dans l'IP classique et utilisable sur des paquets UDP. Contrairement à TLS, DTLS permet la réception des datagrammes dans le mauvais ordre et accepte la perte de paquets. DTLS fournit une solution pour établir et gérer les clés de sessions grâce notamment à un mécanisme de poigné de main. Néanmoins, ce mécanisme ne peut avoir lieu que si des compromis sont faits. Grâce à DTLS, la confidentialité, l'authenticité, l'intégrité, la fraicheur et l'acquittement sont assurés tout en limitant le surcoût dans la taille de la trame. Bien que facile à gérer, DTLS nécessite l'implication du développeur de l'application dans la sécurité.

Enfin, il est possible de mettre en place la sécurité couche MAC. Cela permet de restreindre l'accès au WSN. Peu importe la couche où la sécurité est déployée, la mise en place de mécanisme de sécurité ajoute de la consommation d'énergie en plus et peut donc causer des troubles dans des réseaux contraints en énergie.

6LoWPAN permet donc d'établir les protocoles que la norme IEEE 802.15.4 laissait à la charge des couches hautes tout en permettant l'utilisation de l'IP.

3.3.5 Les protocoles utilisés dans 6LoWPAN

Dans un réseau de capteurs connecté au monde IP classique, il existe trois intervenants différents. Le premier est la *gateway* ou 6LoWPAN Border Router (6BR) qui est le point d'entrée du réseau.

Ce rôle de 6BR doit être assuré par un nœud FFD. Typiquement, cette fonctionnalité peut être assurée par un Raspberry PI branché sur secteur. Le deuxième intervenant est le *root*. Cette fonctionnalité peut être assurée par le 6BR ou confiée à un nœud différent. Néanmoins, celui-ci se devra d'être FFD. Le *root* va permettre au réseau de se créer et de s'organiser autour de lui. Il constitue le sommet du DODAG. C'est lui qui va récupérer les données collectées dans le réseau (nœud puits).

Enfin, le dernier intervenant concerne les capteurs présents dans le réseau. Suivant s'ils sont FFD ou RFD et dans ce cas, ils fonctionneront uniquement sur piles avec des périodes de veilles, ils vont pouvoir assumer deux rôles :
routeurs : vont jouer leur rôle de capteur en collectant les données de leur environnement mais vont également transmettre au *root* les données qu'ils reçoivent des autres nœuds.
feuilles : n'ont pas la capacité de router une trame et vont donc simplement collecter des données et les transmettre.

Les différents intervenants maintenant définis, le réseau peut se mettre en place. Pour cela, RPL détermine un protocole de découverte de voisins. Comme l'expliquent Silva, Leithardt et al. dans [82], il existe trois approches différentes de découverte de voisins utilisables dans les réseaux 6LoWPAN. RPL, de par son fonctionnement, se rapproche du système *ICatchYou*. Dans cette approche, c'est le nouvel arrivant qui va s'annoncer et attendre que le *root* ou un nœud présent dans le réseau réponde. Cette approche réduit le nombre d'échange en *broadcast*. Dans les réseaux IPv6 classique, le protocole correspondant est le Node Discovery Protocol (NDP). Celui-ci va utiliser des trames ICMPv6 afin de transmettre les informations (adresse MAC...) du nouvel arrivant, du réseau déjà déployé et de son fonctionnement.

Lors de l'établissement d'un réseau ou dès lors où un nouvel arrivant veut pouvoir communiquer au sein de ce réseau, c'est à dire, appartenir à ce réseau, ce dernier doit procéder à différents protocoles qui dépendent du type de réseau. Il doit d'abord rejoindre le réseau, c'est ce que l'on appelle le *join*. Durant cette phase, le nouvel arrivant va s'identifier auprès de ses voisins et apprendre la configuration du réseau auquel il souhaite appartenir. Il va ainsi échanger des trames comportant les informations nécessaires à son identification et recevoir en retour les paramètres qu'il doit utiliser (sécurité, préfixe IPv6...). Dès lors où il possède toutes les informations nécessaires, ce dernier va effectuer un protocole d'association indiquant aux voisins qu'il est autorisé à communiquer et par qui il doit communiquer. Afin de s'associer et de communiquer au sein du réseau, le nouvel arrivant doit posséder une adresse *link local* et une adresse globale.

Pour obtenir ces adresses, IPv6 défini deux protocoles de configurations.

- La première est le DHCPv6. Elle correspond au schéma *leader-based* défini par Ghosh et al. dans [83] pour les réseaux ad-hoc. Dans cette approche *stateful*, le nœud va obtenir son adresse par une tierce partie. La création de l'IID (cf. Figure 3.10) repose sur l'intervention d'un serveur pour délivrer des adresses disponibles ou pour fournir les informations nécessaires à l'auto configuration de l'adresse. Un nœud va envoyer une demande d'adresse à tous les serveurs à proximité. Ceux-ci vont renvoyer en réponse, s'ils en ont, une adresse disponible. Le nœud choisie l'une d'entre elle et renvoie un message à chacun indiquant son choix. L'avantage de ce schéma repose sur l'unicité des adresses dans un réseau. En effet, le serveur gère l'attribution des adresses. Cette technique n'est pas facilement réalisable dans un WSN car les nœuds ont besoin de configurer leurs adresses rapidement, même s'ils n'ont pas accès à Internet ou au serveur. Cette fonction ne pouvant pas être réalisée par un nœud contraint, il faut donc assurer aux nœuds un accès à un serveur en permanence. Or, ce n'est pas forcément le cas ni l'utilisation qui est recherchée derrière les WSN. Il est donc préférable que les nœuds puissent auto configurer seuls leurs adresses.

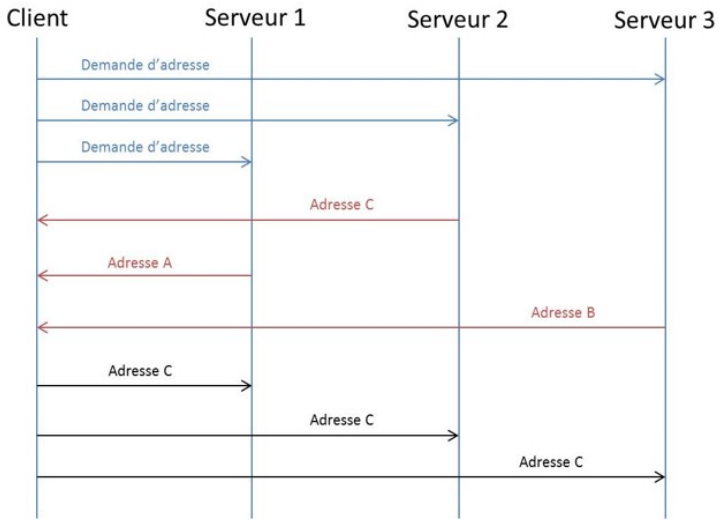


FIGURE 3.10 – Création d'une adresse avec le procédé DHCPv6.

- Pour faciliter la configuration de l'adresse, une méthode appelée StateLess Address AutoConfiguration (SLAAC) [84] a été publiée en 2007. Cette méthode, apparentable au schéma *decentralized* de [83] permet à un nœud souhaitant obtenir une adresse IPv6 d'auto configurer son IID mais également de procéder à la vérification de l'unicité de cette adresse. Pour cela, il va utiliser le mécanisme DAD. C'est une approche *stateless*. La partie IID peut être générée de plusieurs manières (nombre aléatoire, identifiant...). Néanmoins, dans les WSN, afin de faciliter la création de cet IID et de réduire les coûts de calculs, la méthode préconisée va consister à créer cet IID à partir de l'adresse IEEE 802 (cf. Figure 3.11). Dans le cas du 6LoWPAN, cette adresse correspond à l'adresse MAC IEEE 802.15.4. Ce choix permet de réduire également l'espace mémoire nécessaire pour l'auto configuration car cette donnée étant déjà stockée, il ne reste donc qu'à la réutiliser. De plus, le protocole DAD n'est plus nécessaire ce qui réduit la consommation d'énergie. En effet, l'adresse IEEE étant unique, il n'y a que très peu de chance que celle-ci soit déjà utilisée, ce qui réduit le surcoût de trames échangées lors d'un *join*. Pour passer de l'adresse IEEE 802.15.4 de 48 bits à un IID de 64 bits, le nœud va procéder comme indiqué sur la Figure 3.11. Il va tout d'abord prendre son adresse et la scinder en 2 parties égales. Il va ensuite insérer entre ces deux parties les valeurs 0xFF 0xFE lui permettant de correspondre au formalisme des adresses Extended Unique Identifier (EUI)-64. Enfin, il va mettre le bit 7 à 1 indiquant un identifiant local. Si cette adresse correspond à une adresse d'un groupe, il est possible de mettre le bit 8 à 1 indiquant alors une adresse de groupe.

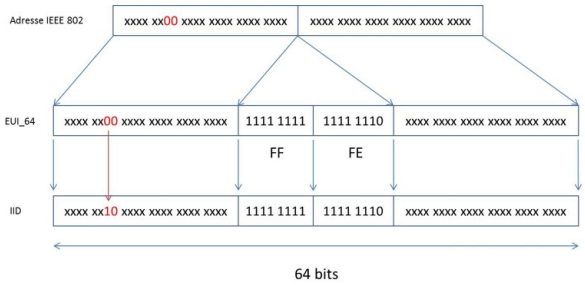


FIGURE 3.11 – Création d'une adresse avec le procédé SLAAC.

Ancillotti, Bruno et al. implémentent dans [85] 3 autres méthodes alternatives d'auto configuration des adresses qui conjuguent approches *stateful* et *stateless*. Les méthodes décrites permettent d'assigner des blocs d'adresses disponibles à des routeurs ou proxy qui vont ensuite jouer le rôle de serveur DHCPv6 pour leurs enfants. Ce fonctionnement est semblable à la version hiérarchisée utilisée dans ZigBee. Ces méthodes

combinent les avantages des deux méthodes précédentes. Néanmoins, ces approches fonctionnent mal avec RPL et contraignent fortement la topologie du réseau et son évolution.

De même, Wang, Sun et al. dans leur article [86] proposent un nouveau schéma de routage optimal qui utilise les voisins 1 ou 2-sauts pour calculer le chemin optimal. Ils partent du constat que dans une topologie en arbre comme RPL, les ressources du *root* mais également de ses voisins sont consommées de façon excessive. Ils définissent alors un système d'adressage *stateful* décentralisé qui permet de changer facilement d'adresse si son parent disparait sans avoir de pertes de communications. L'adresse IPv6 est adressée différemment. La première partie similaire au préfixe indique l'identifiant de l'arbre. Un bit est ensuite utilisé pour indiquer si le nœud est FFD ou RFD. Une partie indique ensuite l'identifiant du nœud uniquement en cas de FFD. Enfin une autre partie permet de faire la même chose mais dans le cas d'un RFD. Ces deux parties sont données par le parent. Le chemin est inséré dans l'en-tête. Ce schéma est sensible à la taille du réseau. Le coût de l'adressage mais également le retard induit augmentent avec la taille du réseau.

Son adresse enfin configurée, un nœud pourra ensuite continuer le protocole d'association. Dans RPL, les protocoles de *join* et d'association sont réunis en un seul. La formation de la topologie avec RPL commence par la désignation d'un nœud comme nœud *root*. L'adresse de ce nœud sera utilisée comme "DODAG ID" permettant d'identifier le DODAG.

Lorsqu'un nœud souhaite rejoindre le réseau déjà formé, ce dernier va envoyer une trame ICMPv6 de type DODAG Information Solicitation (DIS). Grâce à celle-ci, il va avertir ses voisins de sa présence. Ces messages sont utilisés pour solliciter activement des informations sur la topologie. La réception de ce DIS par les routeurs déjà présents ou par le nœud *root* s'il est à portée, provoque le *broadcast* d'une trame ICMPv6 de type DODAG Information Object (DIO) en réponse. Ce DIO contient toutes les informations nécessaires pour joindre le réseau et notamment :

- Le rang. Il permet de mettre à jour la topologie. Ce rang indique la distance de chaque nœud par rapport au nœud *root*. Plus un nœud est éloigné du *root*, plus la valeur de son rang est élevée. A l'inverse, un nœud proche du *root* aura un rang faible mais toujours supérieur à celui du *root*. Lors de la formation du DODAG, un nœud doit d'abord choisir un parent parmi ses voisins. En effet, lorsqu'un nœud souhaite rejoindre le réseau, plusieurs voisins l'avertissent de leurs présences. Il doit alors sélectionner l'un d'entre eux pour faire transiter ses informations jusqu'au nœud *root*. Ensuite, ce nœud doit calculer son rang qui doit obligatoirement être supérieur au rang de son parent pour éviter de créer des boucles dans la topologie. Le calcul du rang d'un nœud ainsi que le choix de son parent préféré dépend de l'Objective Function (OF) utilisée. Ces OF définissent comment les métriques sont calculées. Il existe différents niveaux d'OF. Ces OF sont définis en accord avec les besoins des DODAG. Ainsi RPL permet d'adapter le routage avec les nécessités de certaines applications. La RFC 6551 [87] permet de définir l'OF la mieux adaptée aux besoins des réseaux LoWPAN. Dans ce cas, c'est l'Expected Number of Transmissions (ETX) qui est utilisé. C'est la méthode la plus simple et la moins coûteuse en terme de calcul mais elle n'est pas forcément optimisée. Le choix de la métrique va donc dépendre de l'application désirée et donc de l'efficacité recherchée.
- La méthode utilisée pour la configuration de l'IID. Généralement, cette valeur indique l'utilisation de SLAAC. Le champ laisse néanmoins la possibilité de choisir une autre méthode.
- Le préfixe des adresses IPv6 présentes dans le réseau. Le nouvel arrivant ne possède pour l'instant qu'une adresse *link local* forgée grâce à SLAAC et au préfixe FE80, il lui faut donc obtenir une adresse IP globale afin de communiquer dans le réseau. Comme expliqué, une adresse IPv6 possède deux parties : le préfixe et l'IID. L'IID étant obtenu par SLAAC, il reste uniquement le préfixe à configurer. Ce dernier va donc être configuré avec celui envoyé dans le DIO.

Son adresse configurée et le choix de son parent effectué, le nœud va pouvoir émettre une trame Destination Advertisement Object (DAO) à l'intention de son parent. Dès lors, son adresse va être ajoutée aux tables de routage nécessaires et il va pouvoir communiquer au sein du réseau. Pour ce faire, les nœuds utilisent des trames UDP qu'ils adressent au destinataire via son adresse IP. Les adresses MAC contenues dans les trames, quant à elles, indiquent les adresses source et destination *hop-by-hop* et sont donc modifiées à chaque fois que la trame est routée. Il est donc important de maintenir les tables de routage à jour de façon à connaitre quelle adresse MAC destination remplir pour transmettre la donnée au bon récepteur.

Deux modes de stockage des tables de routage existent. Le premier appelé mode *non storing* ne maintient une table de routage qu'au niveau du nœud *root*. Les autres nœuds ne connaissent l'adresse que de leur parent. Pour communiquer, les données sont remontées jusqu'au nœud *root* qui peut ensuite les retourner au nœud concerné si celui-ci appartient au réseau ou les envoyer à l'extérieur du réseau. Ce schéma apporte un surcoût d'en-tête notamment quand la trame doit être routée dans le réseau car le nœud *root* doit alors ajouter tout le chemin jusqu'à la destination dans la trame. Cette utilisation a donc besoin de plus de puissance et de bande passante. Dans le mode *storing* une table de routage locale est maintenue au niveau des nœuds intermédiaires. Prenons le DODAG de la Figure 3.12. Imaginons que le nœud 8 souhaite communiquer avec le nœud X. Dans le mode *non storing*, la donnée passerait par 7 puis 5 puis 2 pour atteindre le *root* 1 avant de redescendre par 2 puis 6 avec des en-têtes indiquant ce chemin. Dans le mode *storing*, le nœud 8 possède une table de routage. Malheureusement celle-ci ne contient que l'adresse de son parent et pas celle de X. Il va donc envoyer sa trame un rang au-dessus c'est à dire à son parent 7. Quand 7 reçoit la trame, il regarde dans sa table de routage s'il possède le chemin pour atteindre X. Comme ce n'est pas le cas, il va faire la même opération que 8 et l'envoyer à son parent 5. Ainsi de suite jusqu'à atteindre 2. Dès lors, 2 possède dans sa table une route indiquant que pour atteindre X il est nécessaire de passer par 6. Il va donc émettre la trame à 6 qui la fera suivre à X. La trame n'a pas eu à remonter jusqu'au *root* ni à voir ses en-têtes modifiés et la taille de la trame augmenter. En revanche, ce mode demande aux nœuds de stocker des tables plus ou moins importants qui peuvent prendre de la mémoire dans l'environnement contraint. Une table des voisins contenant les adresses MAC des voisins 1-saut est également enregistrée en plus de la table de routage.

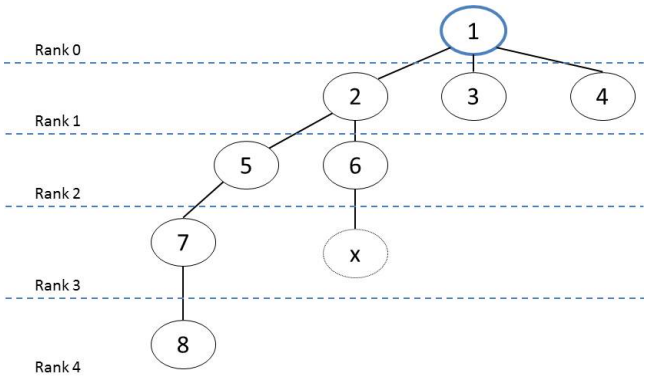


Figure 3.12 – Exemple de DODAG.

RPL défini également un protocole de maintien des tables de routage. Chaque nœud routeur doit envoyer périodiquement une trame DIO afin de connaitre quels sont ses enfants encore présents et s'il y en a des nouveaux. Chaque enfant répond au DIO reçu par un DAO indiquant ainsi qu'il n'a pas bougé et qu'il est toujours présent dans le réseau. Afin de réduire l'impact de cet envoi périodique sur le nombre de trames émises surtout dans le cas de réseaux statiques, RPL permet d'adapter la périodicité aux besoins du réseau. Pour cela, l'algorithme Trickle [88] est utilisé. Son fonctionnement est le suivant : si le message reçu (par exemple le DIO) à un contenu semblable à celui que l'on devait émettre alors on n'envoie pas et on augmente la durée d'attente avant la prochaine émission sinon on émet. Ainsi des nœuds épars pourront transmettre plus que des nœuds où le trafic est dense. L'algorithme définit également une limite haute où l'émission interviendra même en cas de ressemblance.

RPL dispense également les nœuds des protocoles de *rejoin* et de réassociation. En effet, un nœud ayant effectué une première fois le *join* possède dorénavant une adresse IPv6 valide dans tout le réseau. Si celui-ci se déplace ou change de parent, il n'aura pas à changer d'adresse. Il lui suffit donc de répondre au DIO du parent à qui il souhaite être rattaché.

S'appuyant sur RPL, 6LoWPAN et IPv6 facilitent donc la mise en place de réseaux vastes, dynamiques et auto configurés reposant sur plus de flexibilité dans l'établissement et le maintien du réseau. Cette solution a été choisie pour être déployée dans de nombreux OS.

3.3.6 Etats de l'art des systèmes d'exploitation existants

Le déploiement de WSN 6LoWPAN passe par la résolution de nombreux challenges. Le premier concerne le choix de l'OS qui sera déployé.

Les OS traditionnels trop gros pour des nœuds contraints, des OS spécifiques ont vu le jour.

Ces derniers doivent répondre à de nombreux problèmes. Ils doivent pouvoir supporter de nombreuses plateformes ainsi que des technologies de communications hétérogènes. Ces OS doivent pouvoir être utilisés pour la création des *gateway*. Pour cela, ils doivent gérer l'hétérogénéité des couches MAC et Physique et posséder une couche Réseau IP. Ils doivent bien sûr réduire la consommation d'énergie introduite par l'utilisation d'un OS. Etre, autant que faire se peut, temps réel. Et bien sûr permettre la mise en place de protocole de sécurité et des mises à jour du logiciel embarqué.

L'étude réalisée par Hahm, Baccelli et al. dans [89] permet de comparer les OS existants suivant leurs besoins et leurs fonctionnements. Les OS *open source* peuvent être classés en trois catégories :

- Les OS temps réel pur (RTOS). Ces OS sont conçus pour garantir des applications temps réels. Ces OS apportent de grandes contraintes lors du développement d'applications ce qui rends leur portage sur différentes plateformes compliqué. Dans cette catégorie se trouve les OS comme eCos, RTEMS ou encore le plus utilisé FreeRTOS. Ce dernier, créé en 2002, est sous licence GPL et offre uniquement un *kernel open source*. Il ne possède pas de couche Réseau. Ce sont des outils et librairies fournis qui permettent le déploiement des capacités réseaux. L'ordonnancement est préemptif. Il utilise un modèle de programmation basé sur le *multi threading*.
- Les OS *multi-threading*. Ce modèle est semblable à celui utilisé dans les OS comme Linux. Chaque *thread* gère sa couche. Il est donc nécessaire de permettre le switch entre *thread*. Chaque processus peut être interrompu. Néanmoins, ce type de fonctionnement apporte un surcout de mémoire. RIOT créé en 2013 fonctionne avec ce modèle. Il a été conçu pour être simple d'utilisation pour les développeurs. Il possède les couches pour 6LoWPAN mais également pour la version 2012 du standard IEEE 802.15.4. De par sa conception, RIOT est utile pour les réseaux déterministes.
- Les OS *event driven*. Cette approche est celle la plus utilisée. L'ordonnancement est conditionné par des évènements signalés par des interruptions. Ceci permet de réduire l'impact mémoire et la complexité. Contiki OS est le meilleur représentant de cette catégorie. Sous licence BSD, il supporte un grand nombre de plateformes. Sa couche IPv6 est certifiée.

Les *event driven* semblent donc être les OS les plus intéressants car réduisent l'impact mémoire. Ils semblent répondre à un maximum des problèmes auxquels doit faire face un OS léger.

Avec la publication en 2012 de la nouvelle norme IEEE 802.15.4, certains de ces OS tentent d'intégrer cette nouvelle couche. Watteyne, Handziski et al. dans [90] comparent la maturité d'implémentation sur ces différents OS et les challenges qu'il reste. Il n'existe pas réellement d'implémentation complète et efficace. OpenWSN est toutefois plus avancé dans l'implémentation que Contiki où celle-ci est encore en développement. OpenWSN entre dans la catégorie *event driven*. Néanmoins, OpenWSN ne supporte pas le mode *storing* de RPL. Il possède également d'autres limitations pour le déploiement de WSN par rapport à Contiki.

Contiki offre donc un bon potentiel pour devenir un standard référence des WSN.

3.3.7 Contiki OS

Contiki [91] est un OS léger dédié WSN. Il a été pensé pour assurer la connectivité Internet des objets contraints. Il est écrit en langage C et supporté par de nombreuses plateformes. Créé en 2003, il possède une architecture monolithique axée autour d'un noyau. Il fournit également une couche d'abstraction matérielle. Il est prévu pour fonctionner avec des plateformes très contraintes ayant des MCU de 8 bits. Il peut également fonctionner avec des MCU de 16 ou 32 bits.

Il fournit deux piles différentes. La première pile "µIP" (Figure 3.13) permet de mettre en place le modèle OSI définit précédemment pour l'utilisation de 6LoWPAN. Il supporte IPv6 mais également TCP, UDP et

ICMP. La deuxième "RIME", plus légère, fournit un ensemble de couches permettant les communications. Elle est prévue pour un usage dans les WSN.

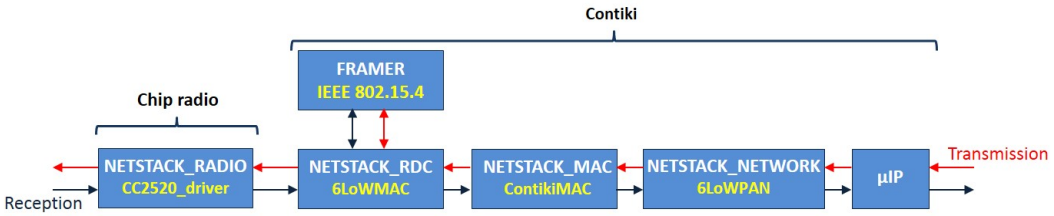


FIGURE 3.13 – Pile µIP de Contiki.

Contiki offre une implémentation de RPL et de l'algorithme Trickle. Enfin, une implémentation logicielle d'un AES est fournie pour sécuriser les communications au niveau de la couche MAC. Contiki utilise les *protothreads* qui permettent de compiler les programmes en évènements afin d'être par la suite gérés par l'ordonnanceur.

Contiki est déployé dans de nombreuses utilisations réelles et utilisé dans des produits commerciaux. Il est également la base de nombreux travaux de recherches et possède donc une grande communauté de contributeurs.

3.3.8 Les nœuds de capteurs utilisés

Le deuxième challenge que doit relever le déploiement de réseaux 6LoWPAN concerne le choix du matériel. Contiki supporte un grand nombre de plateformes. Contrairement à certains OS, il ne limite pas la classe d'appareil utilisée.

En effet, l'IETF a publié une catégorisation des différents appareils pouvant être déployés dans l'IoT. Pour classifier les appareils, l'IETF a utilisé la capacité en mémoire, critère important dans le monde du contraint. Trois catégories ont ainsi émergées :

- Classe 0. Dans cette catégorie se trouvent les appareils les plus contraints en mémoire. C'est dans cette catégorie que l'on retrouve principalement les nœuds qui seront utiles pour les WSN. Ils possèdent beaucoup moins de 10 kB RAM et 100 kB de mémoire flash.
- Classe 1. Cette catégorie regroupe les appareils avec des ressources moyennes autour des valeurs précédentes. C'est généralement ce type de nœuds qui assure des fonctionnalités telles que le routage. Ils ne peuvent implémenter des protocoles comme HTTP mais peuvent utiliser ceux dédiés contraints comme CoAP.
- Classe 2. Ce sont les appareils les moins contraints par rapport aux précédents. Néanmoins, il reste plus contraint que les appareils tels que les *gateway*.

Pour la classe 0, l'utilisation d'un OS est très compliquée mais pas impossible. Les OS doivent optimiser au mieux l'espace mémoire et les allocations. Il est très compliqué de faire fonctionner un OS temps réel dans ce type de nœuds. Contiki peut fonctionner avec les trois classes même si, pour les appareils de classe 2, d'autres OS lui seront préférables.

Pour cette thèse, trois types de carte vont être utilisés. En effet, afin d'analyser les fuites d'informations ayant lieu lors de communications sans fil IEEE 802.15.4, il est nécessaire de mener des expériences sur le modèle OSI complet. Nous avons décidé de comparer les failles existantes avec ZigBee mais également avec 6LoWPAN. Ces études nous permettrons de mettre en place la meilleure sécurité mais également d'identifier les failles communes entre les deux standards.

(a) Carte Waspmote. (b) Carte Wismote. (c) Carte Openmote.

FIGURE 3.14 – Matériels utilisés.

La première carte utilisée est une Waspmote (Figure 3.14(a)). De la marque Libelium, cette carte *open source* possède une RAM de 8 kB et une Flash de 128 kB. Elle appartient à la classe 1. Elle possède également 8 entrées/sorties numériques et de nombreux bus (SPI, UART...) ainsi qu'un capteur de température, d'humidité et un accéléromètre. La *front end* utilisée est une XBee embarquant la version PRO du standard ZigBee permettant les communications sur la bande 2,4 GHz. Cette carte a été utilisée pour l'analyse des fuites d'identité dans un réseau ZigBee. Un *Integrated Development Environment* (IDE) est nécessaire afin de programmer les cartes. Les trois types de clés de sécurité définies par le standard ZigBee sont déployés. Il est possible de déployer un réseau meshé ainsi que de définir les différents rôles. Néanmoins, le rôle de la *gateway* a été réalisé par un Raspberry PI permettant d'adresser les nœuds depuis le web afin d'obtenir un comportement identique à un réseau 6LoWPAN. C'est une plateforme OSGI (Open Services Gateway Initiative) accessible depuis Internet via une adresse IPv4. Elle inclut un serveur HTTP ainsi qu'une interface RESTful et l'API nécessaire pour le front end XBee.

Pour le réseau de test 6LoWPAN, la carte utilisée est une Wismote d'Arago Systems (Figure 3.14(b)). L'antenne est une antenne CC2520 permettant les communications IEEE 802.15.4. Elle possède une RAM de 16kB et une flash de 128 kB, ce qui la classe en catégorie 1. Le CPU est un MSP430. Un capteur de température, un de luminosité ainsi qu'un accéléromètre sont disponibles sur la carte. Wismote autorise l'implémentation de Contiki comme OS ce qui permet la mise en place d'un réseau 6LoWPAN. Pour cela, une machine virtuelle possédant les *Board Support Package* (BSP) nécessaires à la programmation des cartes ainsi que le code source de l'OS est fournie sur le site de Contiki. Néanmoins, la dernière version de Contiki (3.0) n'est pas disponible. Or, cette version corrige des problèmes de sécurité, implémente plus de caractéristiques notamment pour la dernière version de IEEE 802.15.4. Après plusieurs essais afin de mettre en place le BSP sur Contiki3.0, et du fait que les caractéristiques semblaient trop contraintes, le choix de changer de carte s'est imposé.

Nous avons donc utilisé une carte OpenMote (Figure 3.14(c)). Munie d'une antenne CC2538, ce *System on Chip* (SoC) combine un microcontrôleur Cortex M3 de 32-bit et une radio CC2520. Sa RAM est de 32 Kb et elle possède 512 kB de Flash. Elle est donc plus puissante que les précédentes. La carte se programme grâce à une base et une sonde j-link puis fonctionne sur piles.

3.4 Conclusion

De nombreux standards peuvent être déployés dans l'IoT. Le WiFi et le Bluetooth sont deux de ces standards bien connus du grand public. De ce fait, ils ont également été très étudiés en terme de sécurité mais également pour la protection de la vie privée

De plus, les contraintes fortes des WSN combinées au nouveau modèle de communication (faible portée mais durée de vie importante) font que ces deux standards ne sont pas adaptés aux nouvelles applications des WSN.

Un nouveau standard a donc été publié par l'IETF afin de répondre aux nouveaux besoins de ces environnements. L'IEEE 802.15.4 a été spécialement conçu pour des réseaux à faible débit et portée, où les paquets peuvent être perdus et où la consommation d'énergie doit être maitrisée. Pour cela les couches MAC et Physique ont été étudiées afin de réduire à son strict minimum cette consommation. Des spécifications ont

également été données concernant la capacité à déployer la sécurité couche MAC.

Néanmoins, ce nouveau standard ne définit que les couches basses. Deux modèles OSI basés sur IEEE 802.15.4 ont donc été développés, le ZigBee issu d'un consortium d'entreprises et 6LoWPAN standardisé par l'IETF. Ils permettent de gérer les protocoles d'associations mais également de routage.

Contrairement au ZigBee, 6LoWPAN est prévu pour interconnecter le réseau IP et les WSN grâce à la réutilisation des couches hautes classiques du monde IP et à l'ajout d'une couche d'adaptation entre la couche MAC et la couche Réseau permettant de respecter les contraintes de tailles sur la trame MAC IEEE 802.15.4. Il définit RPL comme protocole de routage adapté aux WSN et utilise SLAAC avec l'adresse MAC des nœuds pour l'auto configuration des adresses IPv6 utilisées dans le réseau.

Etant peu connu du grand public et relativement récent, les métadonnées des en-têtes MAC IEEE 802.15.4 visibles par écoute passive ont été très peu étudiées et leurs exploitations sur la vie privée peu analysées.

Le matériel et l'OS ont été choisis de manière à permettre le déploiement de deux réseaux de tests basés sur le standard IEEE 802.15.4 sur lesquels une écoute passive sera menée. Les informations collectées seront ensuite analysées pour extraire des informations de vie privée sur les nœuds et plus globalement sur le réseau et son environnement. Cette analyse complète permettra de déployer une solution de protection des identifiants adaptée à cet environnement contraint.

Partie 4

Exploitation des métadonnées collectées par écoute passive

Dans cette partie, notre objectif est de montrer qu'il est possible d'inférer un maximum d'information sur les réseaux de capteurs IEEE 802.15.4 bien que la sécurité soit activée. Pour cela, nous allons exploiter toutes les informations qui transitent en clair sur l'air, notamment celles contenues dans les en-têtes des messages. Nous allons montrer qu'en interceptant les messages sur une période temporelle suffisamment étendue, il est facile de déduire le protocole d'échange, la topologie du réseau ou encore l'identité, la capacité ou le rôle des capteurs et des routeurs. Ces informations collectées de façon passive, sans intrusion et avec très peu de moyens sont très utiles pour lancer ensuite des attaques actives visant précisément une vulnérabilité du système. Aussi bien la technologie ZigBee que la technologie 6LoWPAN seront expérimentées, avec pour chacune la mise en œuvre d'un dispositif d'interception dédié.

4.1 Introduction

De notre point de vue, l'IEEE 802.15.4 représente le standard le mieux adapté aux contraintes des déploiements de réseaux de capteurs sans fil contraints. Nous avons montré que la protection de la vie privée dans les WSN est difficile à assurer. Le chiffrement a permis de résoudre le problème de confidentialité des données échangées. Néanmoins, un attaquant peut encore effectuer une collecte massive des métadonnées des en-têtes et déduire des informations sensibles de ces métadonnées.

Afin de fournir la contre mesure adéquate, il est tout d'abord nécessaire d'identifier quelles sont ces métadonnées ainsi que les possibles utilisations de celles-ci par un attaquant. Nous avons donc voulu analyser les métadonnées disponibles dans les réseaux IEEE 802.15.4 et exploitables pour inférer des informations de vie privée sur le réseau et ses participants.

Dans la partie précédente, deux modèles basés sur le standard IEEE 802.15.4 ont été introduits. Le standard ZigBee qui permet de créer des réseaux de capteurs meshés. C'est un standard utilisé notamment dans le domaine de l'habitat intelligent grâce à des modules *plug and play* faciles d'utilisation. Et 6LoWPAN qui, quant à lui, permet l'émergence de nouveaux WSN déployés à grande échelle, possédant des nœuds de capteurs pouvant être en mouvement et surveillés depuis un smartphone grâce au protocole IP.

Afin d'analyser les métadonnées disponibles pour un attaquant passif, deux plateformes sécurisées sont donc déployées : une ZigBee et une 6LoWPAN. Des intercepteurs ainsi que des outils d'analyses sont également développés. Nous avons analysé les fuites existantes dans chacun des réseaux indépendamment puis nous avons voulu identifier quelles informations étaient disponibles lorsque la sécurité est activée à la couche MAC.

4.2 Analyse du réseau ZigBee

4.2.1 Le réseau et l'intercepteur déployés

La première plateforme déployée a permis la mise en lumière des fuites d'informations concernant le réseau ZigBee et ses nœuds. Cette plateforme est composée d'un WSN dont les ressources sont accessibles sur Internet via une interface RESTful. Le premier élément ZG est une *SmartGateway* OSGI implémentée sur un Raspberry Pi. Elle assure l'interopérabilité des standards entre l'Internet et le réseau de capteurs ZigBee (cf. Figure 4.1). Elle est accessible via une adresse IPv4. Son rôle va être d'enregistrer la demande de ressource provenant d'Internet et d'interroger le ou les nœuds concernés afin d'obtenir la donnée désirée. Elle va donc devoir traduire la requête HTTP en demande compréhensible par un nœud ZigBee et la router jusqu'au capteur désiré.

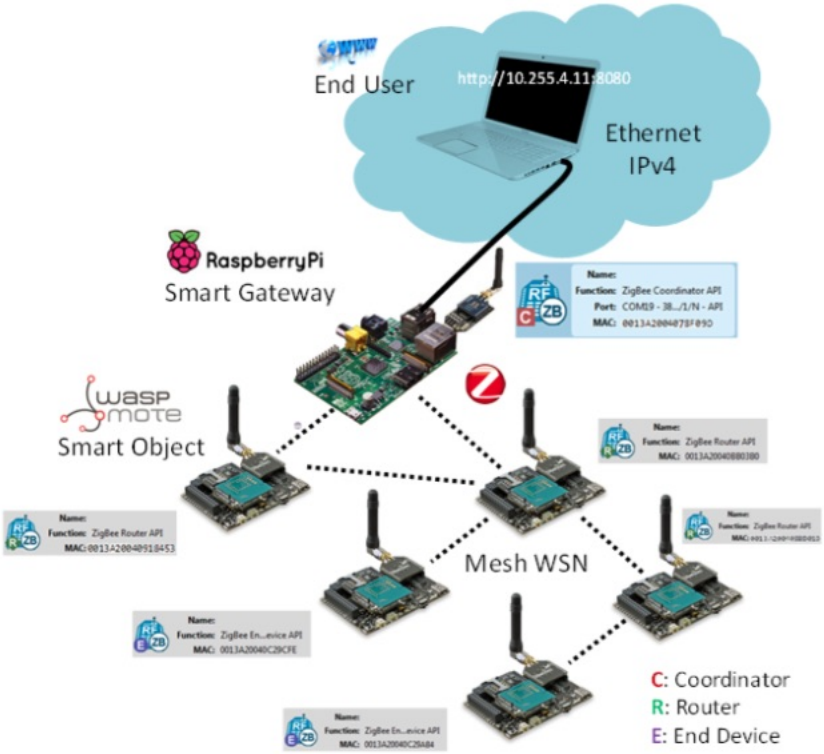


FIGURE 4.1 – Plateforme ZigBee déployée.

Le réseau de capteurs est meshé. Il est formé de nœuds Waspmote équipés d'un *front end* XBee supportant le protocole ZigBee PRO. Parmi les nœuds Waspmote, trois rôles sont présents dans le réseau, selon que le nœud est coordinateur ZC, routeur ZR ou "end device" ZED. Le coordinateur va représenter le contact direct du WSN avec la *gateway*. Comme expliqué dans la partie 3, son rôle va être de créer et de gérer le réseau. Les routeurs permettent de relayer le message depuis l'émetteur vers le destinataire dans un réseau meshé. Enfin les appareils *End Device* sont les feuilles de notre topologie.

Le cas d'utilisation de cette plateforme consiste à permettre à un utilisateur situé dans le domaine de l'Internet ("End User" sur la Figure 4.1) d'avoir accès aux ressources d'un nœud embarquant un capteur de température et un d'humidité. Il envoie alors une requête vers le nœud concerné encapsulée par le protocole HTTP via une URI :
http ://199.166.244.133 :8080/device/device_name/service/service_name/resource/resource_name.

Les routeurs ainsi que les feuilles peuvent également fournir sur demande le niveau de leur batterie. Un utilisateur peut également demander des informations sur les nœuds telles que le nom du fabriquant, le modèle de la carte ou encore le propriétaire.

Pour collecter les informations contenues dans les en-têtes, un intercepteur Killerbee [92] a été utilisé et déployé. Ce *framework* Python *open source* offre un ensemble d'outils permettant le déploiement d'attaques actives de sécurité exploitant les vulnérabilités du ZigBee et du standard IEEE 802.15.4. Ce *framework* peut être utilisé afin de découvrir les failles existantes pour ensuite proposer des contre mesures adaptées. Stelte et al. dans [93] utilisent Killerbee afin de mener une attaque *association flooding* et une attaque par rejeu. Analysant le comportement d'un réseau ZigBee face à ces attaques, ils peuvent ensuite définir un IDS pour celles-ci. Killerbee possède également des outils pour de l'écoute passive. Afin d'exploiter cette capacité pour notre analyse des métadonnées, nous avons utilisé une carte Atmel AVR RZ RAVEN. Néanmoins, d'autres plateformes peuvent être utilisées. Le site de Killerbee [92] donne un tutoriel afin de programmer les différentes plateformes mais également pour l'utilisation des différents outils Python et leurs installations. Afin de programmer la carte Atmel avec le logiciel fourni par Killerbee, du matériel supplémentaire est nécessaire comme une sonde jtag ou des connecteurs.

Au moment de l'expérience, le prix total du matériel nécessaire à la mise en place de l'écoute passive, incluant l'Atmel AVR RZ RAVEN, était environ de 208$. Le temps nécessaire au déploiement de l'intercepteur est d'environ 10 minutes (branchements inclus). La nouvelle méthode décrite sur le site de Killerbee avec le nouveau matériel n'a pas été testée. Néanmoins, le prix du matériel nécessaire a diminué pour atteindre 130$ environ. Le temps mais également l'investissement nécessaire au déploiement est donc faible et accessible au plus grand nombre. Les logiciels sont *open source*. Néanmoins, le logiciel Killerbee est fourni sans possibilité de modifications.

Afin d'exploiter les trames collectées avec Killerbee, celles-ci sont enregistrées automatiquement dans un fichier au format pcap exploitable avec Wireshark. Wireshark [48] est un analyseur de paquets *open source* qui intègre un grand nombre de dissecteurs de standards de communications dont IEEE 802.15.4. Il permet ainsi d'accéder aux différents champs des couches du modèle OSI.

L'intercepteur déployé possède une portée limitée qui restreint les informations collectées. Pour pallier à ce problème, une solution serait de déployer plusieurs intercepteurs au sein du WSN afin de couvrir tout le réseau.
Pour nos expériences, nous avons choisi de ne déployer qu'un seul intercepteur mais de le positionner proche du coordinateur. Il ne collecte que les communications sans fil à sa portée. Placer l'intercepteur proche du coordinateur permet d'avoir une bonne vision des communications ayant lieu dans le réseau et donc des potentielles informations contenues dans les en-têtes. En effet, dans notre cas d'usage, les communications passent forcément par le coordinateur. Nous allons ainsi démontrer que, même avec une vision partielle du réseau sécurisé, les fuites venant des métadonnées sont importantes.

4.2.2 Exploitation des métadonnées du réseau ZigBee sécurisé

4.2.2.1 Description des expériences

Sur la plateforme ZigBee, nous avons mené deux expériences.

Pour la première expérience, les mécanismes de sécurité disponibles avec le *front end* XBee sont mis en place. La clé Réseau définie par le standard ZigBee est fournie avant déploiement au ZC, permettant le chiffrement des paquets au niveau de la couche Réseau. Elle est distribuée ensuite à chaque nouveau nœud dès qu'il a rejoint le réseau. Les en-têtes MAC et ZigBee NTW sont accessibles en clair.
Pour la deuxième expérience, le réseau utilise la sécurité NTW comme précédemment mais également la clé lien qui permet l'envoi chiffré de la clé Réseau. Ces deux expériences permettent de mettre en évidence les métadonnées ainsi cachées par l'utilisation de la sécurité mais également de montrer les fuites restantes permettant d'inférer des informations sur le réseau ZigBee sécurisé.

Le cas d'utilisation commun aux expériences est le suivant.
La *SmartGateway* ZG est tout d'abord connectée au ZC. Les ZR et ZED sont ensuite mis en route les uns à la suite des autres et disposés de façon à former un réseau meshé. Pour cela, certains nœuds sont éloignés du ZC afin de s'assurer qu'ils ne soient pas à sa portée et en conséquence, qu'ils soient obligés de passer par un routeur. Chaque nœud effectue les phases de *join* et d'association qui permet au réseau de s'établir. Dès lors où chaque nœud appartient au réseau, l'échange de données peut débuter. Pour cela, un script est exécuté côté "End User" afin d'accéder aux différentes ressources mises à disposition par tous les nœuds. Il va émettre pour chaque nœud une demande de ressource qu'il enverra à la *SmartGateway* et attendre la réponse. Le coordinateur va alors, pour chaque requête reçue provenant de l'"End User", transmettre la demande au nœud concerné mais également récupérer la valeur et la renvoyer à l'"End User".

Pour chaque expérience, les trames sont interceptées lors de deux phases de vie différentes : établissement du réseau et échanges de données.

Trois analyses des données collectées sont réalisées :

1. Les champs des en-têtes sont analysés indépendamment du type de trames ou du protocole en cours. Les en-têtes de chaque couche OSI ont été analysés en partant de la couche MAC et en remontant jusqu'à la couche Application afin de retrouver la signification de chaque octet et d'en extraire de l'information. Un dictionnaire des différents éléments a ainsi pu être créé.
2. Reconnaissance de protocoles. Les trames collectées sont extraites et groupées pour reconstruire les échanges des différents protocoles. Pour cela, des expressions régulières correspondant à des modèles bien précis sont recherchées. On recherche certain type de trames que l'on sait utilisé pendant ces phases. De par la nature des trames collectées (données brutes), les expressions régulières représentent des champs préalablement identifiés dans l'analyse précédente. Cette analyse donne alors une vision de la dynamique du réseau (nœud entrant/sortant). Il est donc nécessaire de bien connaitre le protocole et la norme afin d'identifier les trames à rechercher pour reconstruire les échanges.
3. Une analyse combinée des deux précédentes est réalisée afin d'identifier des failles de plus haut niveau. On effectue alors une analyse en profondeur des métadonnées des en-têtes des différentes trames d'un protocole complet. Les informations intra couches mais également inter couches OSI sont corrélées. Deux niveaux d'études sont réalisés.
   Le premier consiste à étudier les métadonnées relatives à un seul nœud.
   Le second cherche à établir des relations entre les nœuds du réseau. Pour cela, des règles sont définies afin de reconstruire la topologie mais également afin de découvrir plus d'information sur les nœuds. Ces règles sont nourries par les informations obtenues par la première attaque. Cette analyse donne une vision plus globale que les précédentes sur le réseau.

Ces analyses réalisées sur les données enregistrées peuvent être réalisées en temps réel. Les informations déduites des métadonnées peuvent alors être exploitées de façon préventive comme pour le déploiement d'un IDS ou par un attaquant.

4.2.2.2 Analyses des fuites d'information et exploitations

Dans la première expérience menée avec la sécurité, la clé NTW est envoyée en clair au nœud souhaitant rejoindre le réseau. Cette clé permet de chiffrer les données échangées. Si un attaquant assiste à cet échange, il pourra alors intercepter la clé. Vidgren, Haataja et al. expliquent dans [94] le déroulement de cette attaque ainsi que sa mise en place. Un attaquant peut ainsi déchiffrer les communications et donc espionner tout le trafic. S'il souhaite participer aux communications, il peut alors se légitimer auprès du réseau et grâce à cette clé forger des trames et mener des attaques de sécurité comme l'attaque *sinkhole* et attirer ainsi un maximum de trafic par lui. Il peut ensuite influencer le réseau, la topologie mais également les communications en menant une attaque *selective forwarding*. Dans le cas où l'attaquant est capable de récupérer par simple écoute la clé NTW échangée, l'étude des informations contenues dans les différents en-têtes devient alors identique à une étude d'un réseau sans sécurité déployée. La contre mesure proposée consiste à pré configurer cette clé pour chaque nœud ou à utiliser la clé lien. C'est cette dernière solution qui est testée dans la deuxième expérience. La clé lien est alors pré installée pour chiffrer l'envoi de la clé NTW. Dans cette configuration l'attaquant est limité aux informations des en-têtes non chiffrés par la clé NTW.

Les phases de *join* et d'association n'utilisent que des trames MAC IEEE 802.15.4. Toutes les métadonnées des en-têtes sont donc disponibles par écoute passive même en cas de déploiement de sécurité ZigBee.

D'après notre analyse du standard réalisée à la partie 3, une phase de *join* est initiée par l'envoi d'une trame commande MAC "Beacon Request" comme le montre la figure 3.7 de cette même partie. Tous les ZC et ZR à proximité et appartenant déjà au réseau vont alors répondre en indiquant le PAN ID du réseau dans une trame MAC "Beacon".

La Figure 4.2 montre un exemple de trames analysées lors du *join* avec Wireshark.

```
T Source    Destination   Info
            Broadcast     Beacon Request
  0x0000                  Beacon, Src: 0x0000, EPID: 00:00:00_00:00:00:ff:00
  0x8d17                  Beacon, Src: 0x8d17, EPID: 00:00:00_00:00:00:ff:00

▶ Frame 38: 28 bytes on wire (224 bits), 28 bytes captured (224 bits)
▼ IEEE 802.15.4 Beacon, Src: 0x0000
  ▶ Frame Control Field: 0x8000, Frame Type: Beacon, Destination Addressing Mode: None, Source Addressing Mode: Short/16-bit
    Sequence Number: 148
    Source PAN: 0xbcbc
    Source: 0x0000
  ▼ Superframe SpecificationPAN Coordinator, Association Permit
      .... .... .... 1111 = Beacon Interval: 15
      .... .... 1111 .... = Superframe Interval: 15
      .... 1111 .... .... = Final CAP Slot: 15
      ...0 .... .... .... = Battery Extension: False
      .1.. .... .... .... = PAN Coordinator: True
      1... .... .... .... = Association Permit: True
  ▶ GTS
    Pending Addresses: 0 Short and 0 Long
    FCS: 0xb695 (Correct)
```

FIGURE 4.2 – Protocole de Join ZigBee.

Dans le fichier de trames brutes collectées, deux réponses apparaissent à la suite du "Beacon Request". Cela indique que deux nœuds font déjà partis du réseau et sont à proximité. Il est ainsi possible de commencer une reconstruction partielle de la topologie. Néanmoins, toutes les réponses reçues ne sont pas identiques. Certains champs diffèrent. Cela implique des comportements différents pour les nœuds émetteurs.

Ainsi, dans la Figure 4.2, l'une des trames collectées en réponse indique une adresse source 0x8d17 alors qu'une autre indique 0x0000. Dans la majeure partie des WSN ZigBee, la valeur par défaut de l'adresse du ZC est 0x0000. La norme indique également que pour un ZC les valeurs de "PAN coordinator" et "Association Permit" du champ "SuperFrame Specification" (encadrés en rouge sur la Figure 4.2) sont toutes deux à 1, ce qui est le cas pour l'adresse source 0x0000 contrairement à 0x8d17. Ces deux informations combinées permettent de déduire l'adresse utilisée dans le réseau pour le ZC ainsi que la proximité du nouveau nœud avec le ZC. De plus, comme un ZED ne répond pas à un "Beacon Request", il est possible de déduire que 0x8d17 est un ZR.

Ce travail a été reproduit pour tous nouveaux *join* mais également sur les protocoles d'association menés par les différents nœuds afin d'obtenir des informations sur le réseau et ses participants. Les informations de vie privée inférées ont été enrichies au fur et à mesure des analyses permettant d'identifier les différentes exploitations réalisables grâce à la collecte de ces deux phases.

Les procédures de *join* et association offrent donc de nombreuses fuites d'informations exploitables par un attaquant :

- Un attaquant passif assistant aux phases de *join* et d'association récupère le PAN ID mais également les adresses de tous les nœuds, le type des nœuds, leurs capacités, rôles et sources d'énergie. Il peut déduire les relations entre les nœuds et reconstruire la topologie du réseau. Ces informations peuvent être exploitées pour lancer des attaques DoS.
- Il est possible de lancer des attaques par exhaustion de batterie par attaque *Hello flooding* en forçant un routeur ciblé ou même le coordinateur à répondre à des trames "Beacon Request" et en faisant avorter la suite du protocole.

- Tous les nœuds utilisent le même PAN ID pour identifier le réseau auquel ils appartiennent. Lorsqu'un nouveau nœud souhaite rejoindre un réseau, il doit connaitre ce PAN ID. Il existe deux possibilités. Soit le PAN ID lui a été fourni avant déploiement soit, il doit le découvrir grâce à la trame "Beacon" du *join*. Il sélectionne alors le PAN ID du nœud ayant le meilleur LQI. Dans ce cas, lorsque plusieurs WSN sont présents à proximité, les trames contenant le PAN ID n'étant pas chiffrées ni authentifiées, un nœud n'a pas la possibilité de distinguer quel est le réseau légitime auquel il doit s'associer. Cette situation peut notamment apparaitre dans des immeubles où plusieurs systèmes domotiques coexistent.
- Si un attaquant déploie un coordinateur, il peut tenter d'attirer dans son réseau un maximum de nœuds souhaitant s'associer en renseignant un meilleur LQI que le ZC légitime et en désactivant la sécurité.

La phase d'échange de données comme le protocole de routage utilisent des trames de type ZigBee NTW, le contenu des trames et notamment le chemin contenu dans les trames de routage sont alors chiffrés. Néanmoins, un travail d'analyse de trafic permet de retrouver les chemins de routage qui ne sont plus disponibles en clair et donc de reconstruire la topologie du réseau.

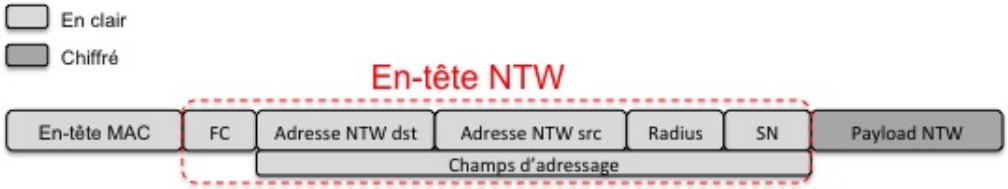


Figure 4.3 – Format simplifié d'une trame ZigBee Network.

En effet, l'en-tête NTW (cf. Figure 4.3) contient les adresses source et destination accessibles en clair afin de permettre le routage. Il contient également un champ appelé "Radius". Il définit une valeur qui est décrémentée à chaque saut de routage. Le "Radius" maximal est commun à tous les nœuds du réseau. En repérant une trame issue du ZC, il est alors facile de retrouver cette valeur maximale.

De plus, la valeur du champ "Sequence Number" (SN) de la couche NTW reste identique tout au long du routage de la trame de l'adresse NTW source à l'adresse NTW destination. En observant le trafic au niveau d'un ZR, il est possible grâce à ce champ d'identifier le routage du paquet. Enfin, l'adresse du prochain saut étant mise à jour pour chaque routeur, un attaquant peut alors suivre le message et en déduire les routes pour accéder aux nœuds. Il connait également l'activité des différents nœuds et leur importance dans les communications.

Dans l'hypothèse où l'attaquant ne serait pas global, les paquets entrant/sortant des ZR aux frontières de la portée de l'intercepteur combinées à la liste des adresses connues du réseau donnent assez d'information pour retrouver la topologie.

Pour conclure, nous avons vu que les phases de *join* et d'association apportent de nombreuses informations de vie privée disponibles même lorsque la sécurité ZigBee est activée. Toutefois, même lorsqu'un attaquant n'assiste pas à ces phases, l'analyse de trafic réalisée lors des phases d'échange de données et de routage lui permet d'inférer de nombreuses informations sur le réseau et ses participants. Cette attaque sera plus longue à mener dans ce cas car il devra s'assurer d'obtenir des communications faisant intervenir chaque nœud du réseau. Néanmoins, de par leur fonctionnement, les nœuds du WSN sont amenés à transmettre leurs données régulièrement à l'"End User" afin d'assurer le rôle de surveillance de l'environnement pour lequel ils sont déployés. Comme l'attaque passive est neutre, l'attaquant peut patiemment attendre que tous les nœuds aient reçu une requête de l'"End User". Il pourra ainsi établir la liste des identifiants des nœuds mais également leur rôle et capacité ainsi que la topologie pour ensuite exploiter ces vulnérabilités dans des attaques actives ciblées. Cette attente, pour le bon fonctionnement du WSN, devrait être largement inférieure à 1 journée.

Regardons maintenant les failles existantes dans un réseau 6LoWPAN.

4.3 Analyse du réseau 6LoWPAN

4.3.1 Le réseau et les outils d'analyse déployés

La deuxième plateforme est un réseau meshé 6LoWPAN. Elle est constituée uniquement de nœuds Wismote embarquant la version 2.6 de l'OS Contiki. Il support le protocole 6LoWPAN et implémente RPL comme protocole de routage et nous utilisons UDP pour le transport des données. Les communications sont multi sauts.

Un nœud assurant le rôle de "Border Router" 6BR a été, contrairement aux autres, branché sur secteur et non mis sur piles. Il représente le point d'entrée/sortie du réseau. Il joue le rôle de puits et collecte donc les données lui parvenant des différents nœuds du réseau. Les données brutes reçues par le 6BR peuvent également être récupérées dans un fichier texte qui pourra ensuite être exploité. Pour cela, ce dernier est branché sur un PC. Il joue également le rôle de nœud *root* et permet la mise en place du réseau et sa configuration. Il fournit également le préfixe utilisé pour la configuration des adresses IPv6 avec le protocole SLAAC.

Les autres nœuds du réseau sont soit des feuilles soit des routeurs. La périodicité du maintien du routage par RPL peut être réglée dans Contiki. Nous avons choisi de garder la valeur par défaut. La durée maximum par défaut entre deux trames RPL est d'environ 17min28s.

Le scénario déployé sur cette plateforme consiste à permettre à chaque nœud du réseau d'envoyer des trames UDP périodiques incluant leur identité mais également les données de leurs accéléromètres.

Pour analyser les communications au sein du réseau 6LoWPAN, un intercepteur Wismote embarquant l'OS Contiki a été développé. Il va permettre de collecter les trames brutes IEEE 802.15.4, en-tête MAC inclus. Des modifications dans l'OS ont été réalisées afin de mettre la radio en mode promiscuité et donc d'éviter le filtrage d'adresses. Seule une carte Wismote et un câble USB sont nécessaire pour la programmation. L'investissement est d'environ 135€ pour le matériel. Pour la programmation, une machine virtuelle *open source* comportant le BSP pour les cartes Wismote est fournie. Elle permet de programmer de manière simple les cartes Wismote. L'un des plus gros investissements concerne la compréhension de Contiki et des fichiers de son noyau. En effet, peu de documentations sont disponibles. Néanmoins, il n'est pas obligatoire de comprendre le fonctionnement complet de Contiki pour déployer l'intercepteur. De plus, dans les versions récentes de Contiki, des exemples de programmes permettant le déploiement d'intercepteurs sont donnés. Il est alors possible de les adapter à la plateforme qui embarquera l'intercepteur. Mis à part la création des programmes des intercepteurs, la programmation de la carte prend environ 3 minutes. Le déploiement reste là encore abordable pour un attaquant.

L'intercepteur Wismote doit permettre deux utilisations.
Soit l'expérience est réalisée puis on analyse après coup les données. On parle alors d'analyse *off-line*. Les données sont collectées et enregistrées dans un fichier qui devra ensuite être transmis à Wireshark pour analyse.
Soit les données sont analysées au fur et à mesure de l'expérience pour une analyse *in-line*.

Néanmoins, afin de traiter et d'interpréter les données, Wireshark a besoin que celles-ci respectent le format pcap. Ce format utilise des en-têtes ajoutés aux données utiles permettant l'appel des bons dissecteurs ainsi que la configuration de Wireshark. Pour assurer nos deux types d'analyses, nous avons donc développés des bridges qui permettent de récupérer les trames brutes MAC collectées dans le réseau et d'ajouter les en-têtes nécessaires au fonctionnement avec Wireshark. Plusieurs bridges ont été nécessaires afin d'assurer les différents types d'analyses.

Par la suite, l'intercepteur Wismote est utilisé en mode *off-line*. De même que pour le réseau ZigBee, nous avons choisi de ne déployer qu'un seul intercepteur et de le placer proche du 6BR.

4.3.2 Exploitation des métadonnées du réseau 6LoWPAN sécurisé

4.3.2.1 Description des expériences

RPL et la couche d'adaptation 6LoWPAN ne permettent pas la mise en place de mécanismes de sécurité efficaces. Le réseau doit alors compter sur les autres couches pour l'activation de la sécurité. Le standard 6LoWPAN prévoit la possibilité de garantir le chiffrement et/ou l'authentification à chaque couche du modèle OSI et donc de moduler la sécurité en fonction du besoin. Il permet la mise en place de la sécurité soit au niveau de la couche transport via l'utilisation de DTLS soit au niveau de la couche réseau grâce à IPsec.

Nous avons donc menés deux expériences. Pour la première expérience, nous avons mis en place DTLS. Les en-têtes jusqu'à la couche transport sont accessibles en clair. Puis dans la seconde nous avons voulu observer les fuites d'information lors de l'utilisation d'IPsec. Contrairement à la première expérience, l'en-tête de la couche transport n'est plus accessible.

Les métadonnées encore disponibles sur le réseau malgré l'utilisation de l'une ou l'autre des sécurités sont étudiées et les vulnérabilités exploitables sont analysées.

Un même cas d'utilisation a été testé.
Lors de la mise en route du WSN, le 6BR est d'abord connecté au PC et alimenté. Les nœuds, déployés dans le bâtiment, sont ensuite mis en route successivement. La portée des Wismote étant faible, un réseau multi sauts est déployé. Dès qu'un nœud a forgé son adresse IP et a rejoint le réseau, il commence à communiquer même si tout le réseau n'est pas encore déployé. L'application est la suivante. Chaque nœud va, toutes les 10s, envoyer au 6BR une trame UDP contenant son identité ainsi que la valeur des trois axes de son accéléromètre. Contrairement au réseau ZigBee, ce n'est pas l'utilisateur qui fait la demande d'une donnée mais les capteurs qui émettent de manière périodique. Ainsi, les deux fonctionnements possibles des WSN sont étudiés.

Pour le réseau 6LoWPAN, les trames interceptées lors de la phase d'échanges de données correspondent donc aux envois périodiques en direction du 6BR. L'intercepteur assiste également à la phase de mise en route du réseau.

Les mêmes techniques d'analyse que celles présentées pour le réseau ZigBee sont utilisées. Néanmoins, avant l'analyse en profondeur des vulnérabilités, une analyse statistique est menée pour identifier le nombre de paquets malformés, ayant un "BAD FCS" ou ne pouvant être analysés. Il est regardé si la sécurité était activée dans les trames collectées. Cette étape permet également d'obtenir la fréquence de chaque type de trames (UDP/ICMP) émises dans le réseau, de les classifier et par conséquence, d'avoir une première vision du fichier et de s'assurer que l'intercepteur a bien collecté les communications.

4.3.2.2 Analyses des fuites d'information et exploitations

Comme pour la plateforme ZigBee, l'intercepteur assiste à la phase d'association des nœuds puis à celle d'échange de trames. Avec RPL, les phases de *join* et d'association sont regroupées en une seule phase. Dans la première expérience, la solution de sécurité activée est DTLS. Nous allons donc analyser les informations disponibles ainsi que leurs exploitations possibles.

DTLS laisse les données des en-têtes RPL visibles en clair. Il est donc possible d'identifier les différents types de trames et donc les protocoles en cours. Les en-têtes des couches plus basses sont également disponibles pour une analyse.

La phase d'association est réalisée grâce à l'utilisation de trames ICMPv6 RPL. La Figure 4.4 montre un exemple d'association repérée dans les trames collectées.

L'envoi d'une trame DIS RPL (en noir dans la Figure 4.4) indique le début de l'association pour le nouveau nœud ayant une adresse MAC terminant par 804. Les adresses sources MAC et IPv6 du nouvel arrivant sont disponibles en clair. Seuls les nœuds routeurs à proximité et appartenant déjà au réseau ainsi que le 6BR vont alors répondre en envoyant un DIO. Il est possible de connaitre les adresses (MAC et IPv6) des trois nœuds émetteurs. Collecter un DIS permet de déduire qu'un nouveau nœud veut rejoindre le réseau et collecter de l'information sur les nœuds à proximité dans le WSN. Néanmoins, à ce stade, il n'est pas possible de distinguer les nœuds routeurs du 6BR. Seul le rôle de feuille peut être écarté. Le nœud adresse

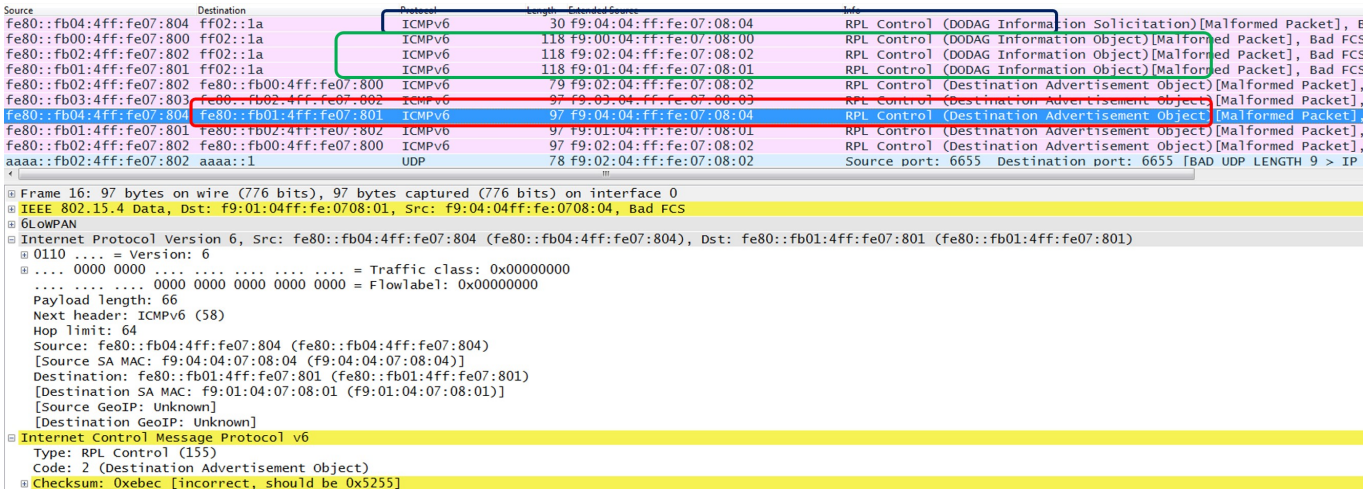


FIGURE 4.4 – Protocole d'association 6LoWPAN.

alors à son parent un DAO RPL. Une topologie locale peut ainsi être reconstruite.

La procédure d'association par RPL offre de nombreuses informations sur le réseau :

- Un attaquant peut facilement identifier le début d'une phase d'association. Il va récupérer les adresses de tous les nœuds. Il peut alors lancer une attaque par DoS afin d'empêcher un nouveau nœud de rejoindre le réseau et le forcer à joindre un autre WSN mais également afin d'épuiser ses ressources.
- Les adresses MAC et IPv6 étudiées avec les DIO émis à la suite d'un DIS permettent de déduire les relations entre les nœuds. Un attaquant peut reconstruire la topologie locale.
- Enfin, l'identification par un attaquant de la phase d'association permet de connaitre le dynamisme du réseau. En effet, si dans un réseau où les communications semblent établies, un nœud vient réaliser une procédure d'association, cela implique que le réseau évolue et n'est pas statique. Cette information peut être pertinente dans certaines attaques.

Afin de maintenir les tables de routage à jour, RPL définit un protocole périodique utilisant des trames DIO et DAO.

Les informations exploitables grâce à l'observation de l'association sont également présentes dans le protocole de maintien des tables. Un attaquant n'ayant pas assisté à la mise en place du réseau pourra donc obtenir les mêmes informations sur le réseau. Il lui suffira de chercher les trames DIO pour les adresses IPv6 observées dans ses données enregistrées ou d'attendre leur capture si c'est une analyse temps réel puis d'intercepter le DAO en réponse. Il pourra donc reconstruire la topologie, identifier les nœuds présents mais également déduire la périodicité des trames RPL. Il pourra enrichir sa topologie en incluant les nœuds feuilles. En effet, lors de l'association, comme ceux ci ne répondent pas aux DIS, les adresses n'apparaissent pas. Dans le protocole de routage tous les nœuds interviennent. Les nœuds feuilles vont donc participer au maintien du routage en envoyant un DAO à leurs parents. L'attaquant pourra récupérer les adresses et construire la topologie.

Cette périodicité de RPL, utile pour le maintien du routage, offre donc un avantage à un attaquant. Il est assuré, même lorsqu'aucunes applications n'est déployées, d'obtenir du trafic et donc de l'information sur le réseau. De plus, ce protocole lui permet de connaitre le dynamisme du réseau. En effet, si lors d'un maintien périodique, un nœud ne communique plus de DIO/DAO, il pourra alors déduire que celui-ci s'est déplacé dans le réseau ou l'a quitté. En cherchant les trames DIO/DAO liées à l'adresse de ce nœud, il pourra alors le suivre et observer ses mouvements dans le réseau.

La traçabilité mais également l'activité des nœuds sont donc facilement accessibles par écoute passive du protocole RPL.

Lors de l'échange de données, celles-ci sont remontées périodiquement au nœud puits, le 6BR. Cet échange utilise les trames UDP. L'intercepteur peut facilement surveiller le trafic arrivant au 6BR. Si l'adresse IPv6 d'un nœud est présent dans un grand nombre de communications ou possède une activité importante, l'attaquant peut alors choisir de mener une attaque DoS sur ce nœud identifié afin de perturber voire d'endommager le WSN. En surveillant les communications, il connait toutes les routes actives ce qui lui permet de reconstruire la topologie sans attendre le protocole RPL. En effet, en analysant les adresses MAC et IPv6, il peut connaitre les nœuds intermédiaires mais également les nœuds source et destination.

En revanche, l'attaquant possède peu d'informations sur les communications en dehors de sa portée. En effet, il connait l'adresse IPv6 source de la trame mais ne possède aucune information sur le nombre de sauts qui séparent ce nœud du premier routeur à sa portée. L'écoute prolongée des communications va lui permettre de reconstruire petit à petit la topologie mais des incertitudes peuvent exister sur la topologie réelle en dehors de sa portée.

L'analyse du réseau protégé par DTLS montre que de nombreuses vulnérabilités viennent des informations de RPL disponibles. La mise en place du chiffrement à la couche Réseau permet de chiffrer l'en-tête de la couche transport indiquant le type de trame (UDP ou RPL) reçue. Un grand nombre d'information contenue dans les trames ICMPv6 et UDP sont ainsi cachées. Un attaquant ne peut plus utiliser les informations disponibles pour identifier le type de trames RPL et donc obtenir des informations sur le protocole en cours. Il ne peut plus, par simple analyse du protocole en cours, déduire le dynamisme du réseau. Il ne peut plus identifier le début du protocole d'association afin de mener ses attaques DoS.

En revanche, les métadonnées des couches plus basses sont toujours émises en clair ce qui offre de nombreuses autres possibilités d'analyses.

La connaissance des adresses déjà présentes dans le réseau combinée à des analyses statistiques (fréquence, temporelle) permet de déduire l'activité des nœuds. Un attaquant peut également découvrir une nouvelle adresse non enregistrée et, par comparaison avec des motifs des communications connus, déduire qu'un nouveau nœud entre dans le réseau. Les trames suivantes sont alors des trames RPL et les adresses collectées correspondent alors aux routeurs ou au 6BR à proximité.

La disparition d'une adresse ou le changement de chemin de routage indique une disparition ou un mouvement de nœud. Enfin, l'auto configuration des adresses par le protocole SLAAC rend les adresses IPv6 fixes même lorsque le nœud se déplace dans le réseau. Cette information peut être utilisée par un attaquant pour déduire le trafic dans le réseau et reconstruire la topologie indépendamment du type de couche transport utilisée (UDP ou ICMPv6), simplement en se basant sur l'en-tête IPv6 qui reste disponible. Il peut déduire l'emplacement dans le WSN des nœuds suivant leurs adresses. Comme l'application consiste à envoyer au nœud puits les données, il peut découvrir l'adresse du 6BR. Il peut ainsi prendre le temps d'agréger des informations sur le réseau et ses nœuds afin de choisir la cible adéquate pour son attaque. Les attaques actives ciblées sont donc plus efficaces et ce malgré la sécurité. Pour conclure, que ce soit avec DTLS ou IPsec, les en-têtes encore visibles offrent de nombreuses informations sur le réseau et ses nœuds. IPsec permet néanmoins de limiter certaines vulnérabilités dues aux informations sur les protocoles disponibles en clair avec DTLS. Toutefois, l'en-tête Réseau contient les adresses IPv6 statiques utilisées pour les communications *end-to-end* et nécessaires pour le routage. Un attaquant peut utiliser ces informations pour de l'analyse de trafic afin de retrouver la topologie du WSN. De plus, l'utilisation de SLAAC, qui permet une auto configuration rapide et simplifiée des adresses IPv6 appropriée aux WSN, facilite le suivi des nœuds lorsque ceux-ci sont en mouvement. Ce fonctionnement facilite également la mise en place d'attaques actives ciblées. Enfin, l'analyse des adresses (MAC et IPv6) combinée à d'autres analyses de vie privée permet de déduire des informations sur le protocole, informations normalement protégées par l'utilisation de la confidentialité à la couche Réseau.

4.4 Analyse des métadonnées du standard IEEE 802.15.4 sécurisé

Les expériences menées sur nos deux plateformes montrent, qu'indépendamment du standard, des fuites d'informations persistent même lorsque la sécurité des couches hautes est activée. Ces informations sur les nœuds et le réseau peuvent être exploitées par un attaquant dans le but de mener des attaques ciblées plus puissantes.

Pour le réseau ZigBee, la mise en place de la sécurité couche NTW est inutile pour les phases de *join* et d'association car les trames échangées sont des trames MAC IEEE 802.15.4. La sécurité des couches plus hautes est donc sans effet. Durant la phase d'échange de données, l'analyse de trafic est faisable même lorsque la sécurité est établie. Pour cela, un attaquant peut utiliser le champ "Radius" combiné aux champs concernant les adresses contenues dans l'en-tête NTW ZigBee. Il peut ainsi reconstruire la topologie et déduire le rôle des nœuds ce qui lui permet de choisir la cible la mieux adaptée pour une attaque active. Une solution consiste à activer la sécurité MAC IEEE 802.15.4. Ainsi, elle permet d'obtenir une phase d'association sécurisée auprès d'un coordinateur légitime car possédant la bonne clé de chiffrement. Les données contenues dans l'en-tête NTW et utilisées pour l'analyse de trafic ne sont alors plus accessibles en clair.

Pour le réseau 6LoWPAN, même si l'utilisation du chiffrement couche Réseau permet de protéger l'association, l'analyse de trafic apporte de nombreuses informations de vie privée sur le réseau et ses participants. La couche Réseau IPv6 est notamment une couche porteuse d'un grand nombre d'informations exploitables par un attaquant pour cette analyse de trafic. Il va alors utiliser les adresses IPv6 source et destination contenues dans l'en-tête Réseau pour suivre le message et recréer les chemins et la topologie. SLAAC permet aux adresses d'être statiques. Un attaquant peut donc également exploiter cette propriété pour suivre un nœud et son activité. Là encore, une solution consiste à activer la sécurité MAC qui assure ainsi la confidentialité des informations portées par la couche Réseau et les couches au-dessus.

Dans les deux réseaux étudiés, l'une des vulnérabilités communes vient de la possibilité grâce à la couche Réseau (NTW ou IPv6) de réaliser de l'analyse de trafic dévoilant des informations sur le réseau, sa topologie et ses nœuds. L'idée est donc d'assurer la confidentialité des métadonnées de cette couche via l'utilisation de la sécurité au niveau le plus bas, c'est-à-dire au niveau MAC.

Le *front-end* XBee dans sa version actuelle utilisée sur les nœuds Waspmote et assurant le déploiement du standard ZigBee PRO ne permet pas d'activer la sécurité IEEE 802.15.4 MAC. Il est donc nécessaire de développer une nouvelle puce radio compatible ZigBee incluant les spécifications du standard IEEE 802.15.4e pour la sécurité. En réalisant les calculs cryptographiques au niveau matériel, l'utilisation de la sécurité devient transparente pour les couches hautes du standard ZigBee. Pour le réseau 6LoWPAN, Contiki permet de mettre en place la sécurité MAC grâce à un AES logiciel. Néanmoins, la version 2.6 de Contiki n'offre pas la possibilité d'activer la sécurité de la version 2012.

Sokullu, Korkmaz et al. montrent dans [95] que la sécurité à cette couche permet également d'empêcher ou de limiter un grand nombre d'attaques actives sur la couche MAC comme les attaques par injection de paquets. Néanmoins, il est nécessaire, à chaque saut, de déchiffrer puis de chiffrer à nouveau une trame lorsque celle-ci doit être routée. De plus, afin d'éviter les failles de sécurité, le management des clés doit être fait avec précaution. Nous avons vu dans l'analyse précédente sur le réseau ZigBee que la clé utilisée pour la sécurité NTW pouvait être envoyée en clair lors de la phase d'association. Ce fonctionnement dangereux est à proscrire au prix de casser la sécurité et la protection de la vie privée du WSN. Une manière correcte de procéder consiste à pré installer en priorité la clé nécessaire pour le chiffrement couche MAC. Celle-ci permet de protéger le réseau d'un attaquant passif extérieur. De plus, elle serait active dès la phase de *join* afin d'authentifier et de chiffrer les échanges. Si le WSN souhaite par la suite mettre en place une sécurité sur les couches plus hautes, les clés pourront être négociées et échangées de manières chiffrées par la clé MAC. Elles peuvent également être pré installées de la même manière que la clé MAC pour éviter qu'elles soient connues de tous les nœuds pré configurés avec la même clé MAC. Néanmoins, cette solution nécessite une réflexion avant déploiement et un travail de configuration par l'utilisateur.

Dans un réseau sécurisé au niveau de la couche MAC, les métadonnées de l'en-tête MAC sont encore exposées à de l'écoute passive. Il est donc nécessaire d'étudier les métadonnées exploitables dans cette couche MAC.

La Figure 4.5 donne les différents champs accessibles à un attaquant lorsque la sécurité est activée à la couche MAC.

Le champ "Frame Control" donne de nombreuses indications à un attaquant concernant le réseau et son dynamisme. Elles lui permettent de choisir l'attaque la mieux adaptée. Notamment, le champ "Security Enabled" (SE) indique si la sécurité est activée. Un attaquant peut utiliser cette information pour choisir un réseau non sécurisé s'il souhaite mener une attaque rapide et peu coûteuse ou encore pour mettre en place les

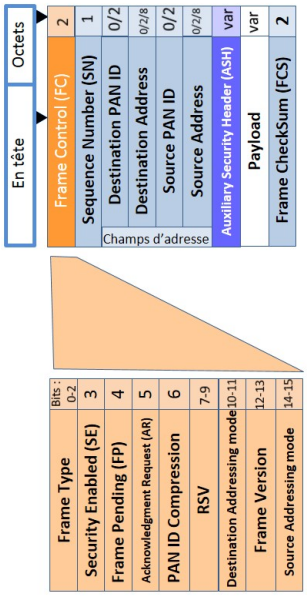


FIGURE 4.5 – En-tête MAC disponible en clair par écoute passive.

outils et attaques appropriées pour tenter de passer la sécurité. Il peut également utiliser la connaissance de l'activation de la sécurité pour mener une attaque par exhaustion de batterie. En effet, lorsqu'un nœud reçoit une trame sécurisée, le standard MAC définit que la trame doit être dans l'ordre déchiffrée puis authentifiée. Un attaquant peut forger une trame ayant les bons champs de l'en-tête MAC mais un *payload* chiffré et un MIC faux. Le nœud va alors faire les opérations coûteuses de vérification inutilement. En répétant l'opération, le nœud va épuiser rapidement sa batterie.

La plus grande faille lorsque la sécurité est activée au niveau de la couche MAC est portée par les adresses. En effet, celles-ci ne peuvent être cachées car utiles au routage. Il est possible de réaliser alors de l'analyse de trafic *hop-by-hop* et ainsi de déduire la topologie. Néanmoins, cette reconstruction est fortement limitée à la portée du collecteur. Elles peuvent également être utilisées pour des attaques ciblées. En effet, ces adresses sont statiques et liées au matériel. Un attaquant peut donc les utiliser pour faire du scan d'adresses, corréler les activités d'un nœud au sein du réseau ou même lorsque celui-ci est mobile entre plusieurs réseaux. La connaissance de l'adresse MAC liée au matériel lui permet d'identifier les vulnérabilités spécifiques à ce matériel et de mettre en place une attaque en conséquence. Les adresses étant statiques, l'attaquant possède tout le temps nécessaire pour mener à bien son attaque.

Afin d'éviter ou du moins de limiter l'utilisation de la connaissance des adresses dans des attaques plus puissantes, une contre mesure consiste à empêcher un attaquant d'observer une adresse et de la relier à un nœud. Pour cela, il est possible d'utiliser des techniques d'anonymat ou des pseudonymes. Ainsi l'attaquant ne pourra pas utiliser l'adresse pour cibler un nœud intéressant ou pour le suivre lorsque celui-ci est en mouvement.

4.5 Conclusion

Dans cette partie, nous avons étudié les fuites d'informations concernant deux modèles basés sur le standard IEEE 802.15.4 : le ZigBee et le 6LoWPAN.

Pour chacun de ces réseaux, nous avons testé différents mécanismes de sécurité proposés et analysé les métadonnées encore disponibles. Les réseaux ont été étudiés pendant deux phases de vie différentes. Dans un premier temps, nous avons observé les informations disponibles lorsque le réseau s'établit. Puis nous avons étudié les communications lors de l'échange de données et du maintien du routage.

Malgré la possibilité qu'offre les deux standards d'activer la sécurité à la couche Réseau, les métadonnées disponibles ainsi que leur exploitation ne sont pas identiques.

Tout d'abord, les phases de *join* et d'association définies par le standard ZigBee sont plus sensibles à de l'écoute passive que celles du standard 6LoWPAN. En effet, la sécurité ZigBee ne permet pas d'assurer la confidentialité de cet échange. L'échange est accessible en clair et un attaquant peut mener des attaques sur le protocole plus facilement. Il peut forger de faux protocoles de *join* et d'association pour mener des attaques DoS ou tenter d'attirer les nouveaux nœuds dans son propre réseau. Il peut également facilement identifier le début du protocole et en déduire la dynamique du réseau. La sécurité couche Réseau de 6LoWPAN permet de cacher cet échange et donc de le protéger contre ces attaques. L'échange étant chiffré, un attaquant ne peut alors pas forger de fausses trames afin de reproduire une fausse association sans connaitre la clé. Un attaquant doit déployer des techniques d'analyse de trafic plus poussées pour identifier le protocole dans les données collectées. Néanmoins, avec des recherches sur les motifs des protocoles et des en-têtes des trames, il est capable d'inférer le déroulement du protocole d'association et donc de l'analyser. Il peut ainsi avoir accès au dynamisme du réseau.
De plus, un attaquant a accès à plus d'informations sur les nœuds et le réseau avec ZigBee. Les champs présents lui permettent de connaitre le PAN ID, les différentes adresses, les capacités des nœuds mais également leurs rôles et, si le ZC est présent, de l'identifier par rapport aux autres nœuds. Il peut reconstruire la topologie. L'écoute du protocole d'association de 6LoWPAN uniquement ne permet pas de distinguer les nœuds routeurs du nœud 6BR. Les informations présentes permettent uniquement d'écarter le rôle de feuille. Comme pour le réseau ZigBee, un attaquant connait le PAN ID et les différentes adresses (MAC et IPv6) et peut commencer une reconstruction de la topologie.

L'échange de données et le maintien du routage permettent à un attaquant, indépendamment de la plateforme, de réaliser de l'analyse de trafic. Cette analyse de trafic permet de retrouver les identités des différents nœuds du réseau et la topologie. Il permet également de retrouver la mobilité des nœuds et leurs activités. Néanmoins, l'en-tête Réseau de ZigBee permet une analyse plus poussée et une reconstruction plus optimale en cas de portée limitée de l'intercepteur. En effet, la connaissance du champ "Radius" accessible en clair dans l'en-tête NTW permet d'indiquer à un attaquant le nombre de sauts réalisés depuis l'émission de la trame et donc, si un attaquant patiente assez longtemps, permet avec la connaissance des adresses de retrouver la topologie hors portée. Dans le cas du réseau 6LoWPAN, le nombre de sauts hors portée n'est pas indiqué. L'attaquant a donc une vision plus limitée et doit déployer plus de force pour reconstruire la topologie complète (par exemple, déployer plusieurs intercepteurs). En revanche, l'utilisation dans les réseaux 6LoWPAN du protocole SLAAC offre la possibilité de suivre plus facilement un nœud même lorsque celui-ci change de réseau. L'adresse MAC étant liée à l'adresse IPv6 un attaquant peut donc déduire des adresses IPv6 les adresses MAC, même des nœuds hors de portée, ce que ne peut pas faire un attaquant dans un réseau ZigBee. Enfin, la présence de l'adresse destination dans l'en-tête réseau permet d'identifier l'adresse IPv6 du 6BR.

Les failles viennent donc principalement des protocoles de *join* et d'association non sécurisés et de l'analyse de trafic réalisable grâce aux différentes informations contenues dans les en-têtes Réseau.

Un moyen de réduire les métadonnées disponibles est de mettre en place la sécurité sur la couche la plus basse du modèle OSI c'est à dire la couche MAC. Ainsi les adresses Réseaux par exemple ne pourront plus être accessibles par simple écoute. La confidentialité des données de la couche Réseau sera assurée. Néanmoins, même lorsque celle-ci est utilisée, l'en-tête MAC reste disponible en clair et sujet à de l'analyse de trafic. Une analyse des métadonnées disponibles lorsque la sécurité MAC est activée a révélé que les adresses MAC utiles pour le routage et ne pouvant être cachées représente une information importante pour l'analyse de trafic.

Les identifiants permanents que sont les adresses MAC peuvent être utilisés pour suivre un nœud ou pour inférer des informations sensibles utiles aux attaques actives ciblées. Un moyen de contrer ces attaques est de masquer les adresses utilisées pour le routage et présentes dans les en-têtes. Ainsi, il ne sera plus possible d'associer une adresse avec un hôte spécifique ou de retrouver sa localisation. Les techniques d'anonymat et d'utilisation de pseudonymes ont alors rapidement été adoptées pour répondre à ce problème. Par exemple, elles sont utilisées dans des produits grands publics comme iOS 8 ou encore Tail Linux.

Néanmoins, ces techniques ajoutent de la complexité lors du routage notamment dans les réseaux contraints. Nous allons donc voir dans la prochaine partie quelles solutions de l'état de l'art existent et si elles sont applicables aux contraintes des WSN.

Partie 5

Solutions de l'état de l'art pour dissimuler ses identifiants

Dans cette partie, les contre mesures de l'état de l'art sont étudiées afin d'identifier la solution de dissimulation des identifiants la plus appropriée aux contraintes des WSN. Nous allons présenter les solutions utilisant les techniques d'anonymat ainsi que celles utilisant les pseudonymes. Nous allons analyser chaque solution vis-à-vis de sa compatibilité avec les contraintes des WSN. Nous allons montrer que beaucoup de solutions dédiées aux réseaux classiques, et notamment les techniques permettant l'anonymat, sont incompatibles et qu'il est nécessaire de déployer une solution pensée pour les réseaux de capteurs.

5.1 Introduction

L'utilisation des identifiants permanents comme adresses ou partie des adresses rend le déploiement des réseaux sans fil plus rapide et efficace. C'est pourquoi les méthodes *stateless* comme SLAAC ont pris de plus en plus de place dans les processus de génération d'adresses. Grâce à la réutilisation de l'adresse MAC liée au matériel, le processus de génération des adresses IPv6 est plus rapide mais assure également l'unicité des adresses sans ajout de trafic.

Ces adresses sont fondamentales pour assurer les communications mais peuvent être exploitées par un attaquant pour mener des attaques ciblées. De plus, les cacher d'une écoute passive n'est pas facilement réalisable. Ainsi, même lorsque le chiffrement est activé à la couche la plus basse, les adresses MAC sont toujours disponibles en clair. Afin de limiter les fuites d'identifiants permanents dans les communications sans fil et leurs exploitations par un attaquant, il est donc nécessaire de déployer une solution de protection de la vie privée. Plusieurs voies peuvent être explorées.

Les solutions étudiées sont basées sur deux principes différents :

1. L'anonymat
2. L'utilisation de pseudonymes

Kelly et al. expliquent dans leur livre [96] que le principe d'anonymat consiste à empêcher une identité d'être liée à un ensemble spécifique d'activités grâce à des méthodes d'obfuscation. Ainsi, dans l'Internet classique, l'anonymat consiste à permettre l'utilisation des services Internet sans révéler l'identité ou la localisation de l'utilisateur. Cette identité inclut les adresses MAC et/ou IP. Un service d'anonymat doit laisser à l'utilisateur le choix de diffuser ou non les informations permettant son identification. De nombreuses recherches ont été menées pour assurer l'anonymat dans le monde IP classique. Kelly et al. dans [96] détaillent le nombre conséquent de solutions existantes dans le monde du filaire ainsi que les avantages et les inconvénients de chaque solution. En ce qui concerne les réseaux sans fil, peu de littérature existe. L'article détaille néanmoins certaines des contre mesures adaptées aux contraintes de ce type de réseau.

En ce qui concerne l'utilisation de pseudonymes, les méthodes utilisées vont générer des pseudonymes que les clients vont pouvoir utiliser pour remplacer les champs d'identités et d'adresses présents dans les paquets transmis pour leurs communications. L'activité d'un pseudonyme ne sera pas obfusquée mais il sera

impossible de remonter à l'identité, l'activité ou la localisation du nœud réel présent dans le réseau. Il sera également impossible de lier deux pseudonymes utilisés par la même entité.

Néanmoins, les réseaux dans lesquels ces solutions sont déployées doivent conserver des performances raisonnables et permettre une qualité de service correcte. Les solutions ne doivent pas non plus introduire des failles de sécurité. Pour qu'une solution de dissimulation des identifiants soit convenable il faut qu'elle puisse fonctionner avec les critères de sécurité du réseau comme, par exemple, autoriser le contrôle d'accès.

Les solutions présentées par la suite vont donc être analysées sur plusieurs points :

- Protection de la vie privée : les attaques favorisées par l'utilisation des identifiants permanents sont-elles encore réalisables ? Des fuites d'identifiants peuvent-elles encore mener à des vulnérabilités exploitables par un attaquant ?

- Environnement : est-ce que cette solution est réalisable dans un réseau de capteurs (nœuds contraints, mobiles, topologie complexe) ?

- Sécurité : est-ce que la solution ne remet pas en cause la sécurité ? Est-il possible de légitimer le nœud avec l'aide / sans aide d'une autorité ?

- Performance : dans quelle mesure la solution détériore les performances du réseau ?

- Concordance : est-ce que cette solution fonctionne avec les protocoles utilisés dans 6LoWPAN (RPL, SLAAC) ? En effet, les solutions décrites par Kelly et al. dans le livre [96] ne sont pas étudiées vis-à-vis d'un réseau spécifique et donc du contexte décrit précédemment dans la partie 3 pour les WSN. Il faut donc veiller à ce qu'une solution soit utilisable avec les protocoles déployés dans les WSN.

5.2 Les techniques d'anonymat

Certaines solutions de l'état de l'art proposent de protéger uniquement l'identité de l'émetteur. Un attaquant ne sera alors plus capable de lier un message avec une source précise. Ces solutions sont intéressantes dans le cas où l'attaquant cherche à obtenir la localisation d'un évènement capté. Il est également possible d'assurer l'anonymat du récepteur. Dans ce cas-là, l'attaquant ne pourra pas connaitre l'adresse du destinataire du message. Ces solutions permettent de protéger les nœuds puits qui sont des éléments importants du réseau. Enfin, une solution plus complète permet l'anonymat de la communication. Un attaquant ne peut alors établir de liens entre l'émetteur et le récepteur d'une communication. Lorsqu'un attaquant ne peut reconstruire la topologie d'un réseau on parle d'anonymat de localisation.

Enfin, un critère fort concerne l'inobservabilité. Un attaquant peut alors soit ne pas être en mesure d'observer les communications, soit ne pas réussir à distinguer une communication d'une autre. L'inobservabilité implique l'anonymat. En revanche, il est nécessaire d'ajouter des techniques de génération de faux trafic aux techniques d'anonymat pour atteindre l'inobservabilité. Ce critère est donc un critère fort et souvent compliqué à mettre en place car nécessite des techniques trop gourmandes notamment pour être déployées dans les réseaux sans fil contraints.

La grande majorité des solutions proposées dans la littérature sont basées sur l'utilisation de techniques de mixage comme celles présentées dans la partie 2. L'idée étant d'altérer l'apparence et/ou le trafic des messages émis dans le réseau afin de protéger la source et la destination du message.

La modification du trafic permet d'éviter les attaques par analyses temporelles ou statistiques en ajoutant du retard et du ré-ordonnancement dans les messages au niveau d'un nœud routeur appelé mixeur. En ce qui concerne la modification de l'apparence d'un message, du chiffrement combiné à du bourrage (*padding*) est appliqué au message de départ.

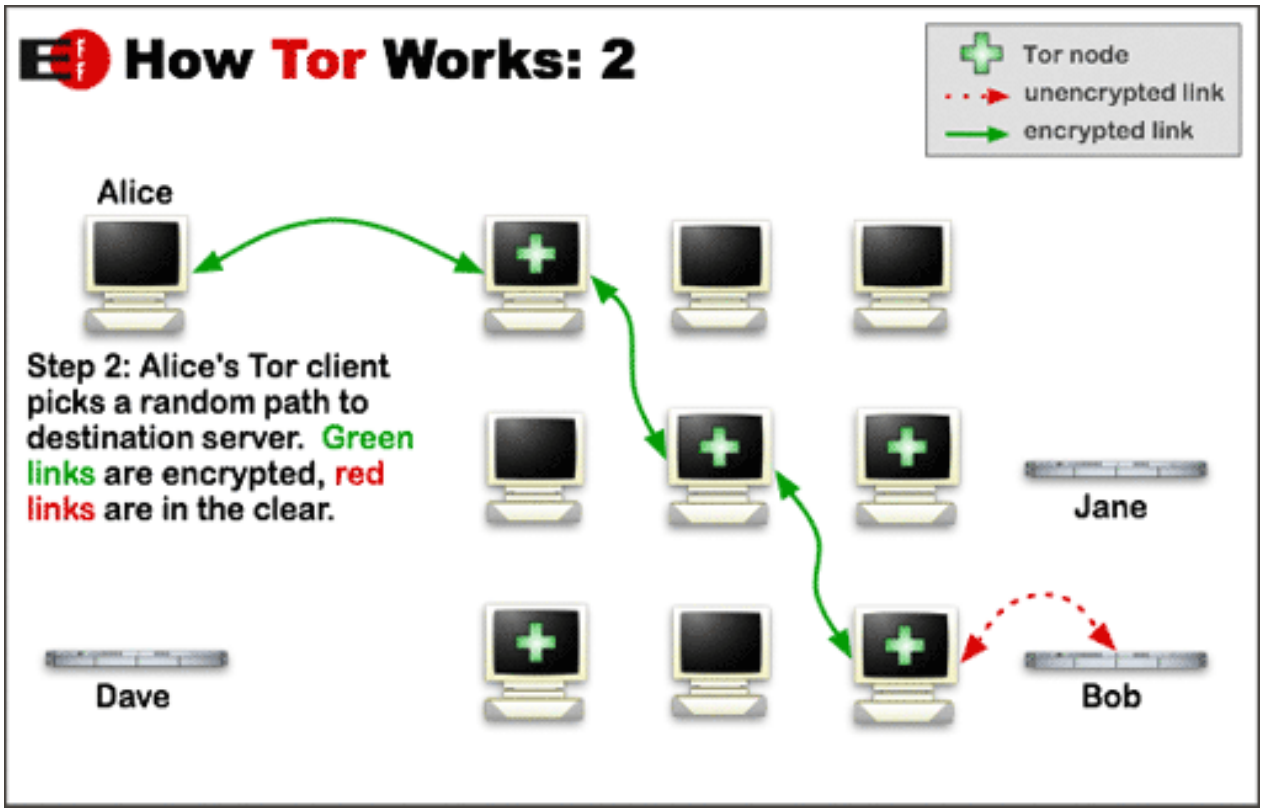


FIGURE 5.1 – Anonymat par TOR.

Dingledine, Mathewson et al. présentent dans [97] The Onion Routing (TOR), deuxième génération de solution utilisant le routage oignon. TOR propose de rendre le routage anonyme de façon à lutter contre l'analyse des en-têtes utilisés pour le transport des données sur Internet. Il assure l'anonymat de l'émetteur, du récepteur mais également des communications contre un attaquant interne ou externe. TOR a été créé pour fonctionner avec des communications TCP. TOR utilise le principe des réseaux mélangés (ou mixeur) appliqué à la couche transport. Cette méthode permet l'utilisation d'un chemin aléatoire à travers des routeurs n'ayant pas une vision complète du chemin.

Sur le site TOR Project [98], le principe de fonctionnement de TOR est expliqué. La Figure 5.1 est également extraite du même site. Lorsqu'Alice va vouloir communiquer avec Bob, elle va tout d'abord interroger un serveur TOR afin d'obtenir la liste des routeurs TOR (désignés par une croix dans la Figure 5.1). Alice va ensuite choisir un chemin aléatoire jusqu'à Bob à travers ces routeurs. Le chemin obtenu ne passe que par 3 des 5 routeurs TOR disponibles. Afin de s'assurer qu'aucun nœud ne connait le chemin complet, Alice va établir des connexions chiffrées différentes entre chaque routeur. Chaque routeur intermédiaire ne connait ainsi que l'adresse du saut précédent et celle du saut suivant. Pour cela, Alice va récupérer de chacun de ces nœuds une clé de chiffrement différente. Elle va ensuite chiffrer, de manière successive, sa trame avec les clés des différents routeurs du chemin. Dans la dernière étape, les routeurs TOR intermédiaires vont tour à tour "éplucher", en déchiffrant grâce à leur clé, la trame afin de connaitre la destination pour le prochain saut. Seule la couche propre au routeur est enlevée. Le chemin se construit au fur et à mesure des déchiffrements. Seule la communication entre le dernier routeur TOR et Bob n'est pas chiffrée.

Lorsqu'Alice souhaite communiquer avec une autre destination, elle doit alors répéter l'opération et choisir un nouveau chemin aléatoire. De plus, le chemin établi n'est valable que pour 10 minutes. Après cela, il est nécessaire de recommencer le processus.

En terme de protection de la vie privée, la construction du chemin se faisant de façon incrémentale, cette méthode est résistante à l'analyse des en-têtes ou aux attaques MITM. Néanmoins, cette technique

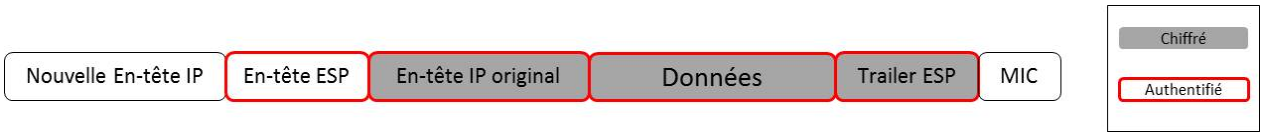


FIGURE 5.2 – IPsec en mode tunnel.

est sensible à de l'analyse de trafic basée sur la corrélation temporelle. De plus, Pries, Yu et al. montrent dans [99] que TOR est vulnérable à des attaques par rejeu permettant de casser l'anonymat et donc de lier une source à sa destination.

Dingledine, Mathewson et al. expliquent dans [100] que le déploiement et le maintien de réseau distribué, anonyme et de faible latence comportent d'énormes challenges. Ces challenges sont encore plus importants lorsque le déploiement concerne un réseau de capteurs contraints où TCP n'est pas le protocole préconisé. De plus, du fait que les adresses soient anonymes, il est compliqué voire impossible de faire du contrôle d'accès. L'un des inconvénients de cette méthode est le choix du chemin qui n'est pas forcément optimale et nuit fortement aux performances du réseau. En effet, afin d'empêcher l'analyse de trafic, le chemin n'est pas direct mais aléatoire. Il peut choisir plus de routeurs que ne le ferait la route directe. Cela introduit des latences mais également un surcoût dans le routage de la donnée qui, dans les réseaux de capteurs, entrainent une surconsommation de l'énergie. Enfin, la taille des champs supplémentaires qu'introduit TOR pour fonctionner induit un surcoût considérable. En effet, TOR fixe 500 octets pour ajouter les en-têtes chiffrés nécessaire à l'anonymat. Ainsi, un attaquant ne peut pas utiliser la taille de la trame pour déduire le nombre de routeurs total mais également le nombre de routeurs restants. Néanmoins, ces en-têtes additionnels sont inutilisables dans un réseau WSN où la taille maximale d'une trame est de 127 octets. TOR est donc une technique d'anonymat intéressante pour les réseaux classiques mais n'est pas utilisable dans les WSN.

Afin de réduire le surcoût apporté par l'anonymat, Matos, Sargento et al. dans [101] proposent d'utiliser les extensions de champs de l'en-tête IPv6 pour transmettre les informations d'adresses réduisant le surcoût à 48 octets. Leur méthode utilise des *waypoints* qui jouent le même rôle que les routeurs TOR. L'inconvénient de leur méthode est qu'elle repose sur un ensemble de contrôleurs qui fournissent et assurent la dissimulation comme par exemple dans le choix du chemin. De plus, un surcoût de 48 octets est encore trop important pour être embarqué dans des réseaux contraints IEEE 802.15.4.

Pour s'assurer de l'anonymat des communications, une autre solution est d'utiliser IPsec en mode tunnel (cf. Figure 5.2). Dans ce mode, l'en-tête IP est chiffré et encapsulé dans un nouvel en-tête IP. Cette méthode, similaire à la création d'un VPN, permet le masquage d'adresses. Il est généralement utilisé entre deux passerelles sécurisées donnant accès à deux réseaux. L'ajout d'un nouvel en-tête IP augmente la taille de la trame à transmettre. En effet, l'en-tête IPv6 possède une taille de 40 octets. L'utilisation d'un nouvel en-tête ajoute donc 40 nouveaux octets. De même que pour les deux méthodes précédentes, IPsec apporte donc un surcoût non négligeable qui le rend inadapté aux WSN. De plus, avec IPsec, les adresses MAC sont toujours disponibles en clair pour de l'analyse de trafic *hop-by-hop*.

Ces solutions montrent que vouloir cacher le chemin est très complexe dans les réseaux de capteurs et souvent trop gourmand en en-têtes et en mémoire nécessaires au stockage des clés interdisant leurs utilisations. De plus, les protocoles de routage ont besoin de flexibilité pour s'adapter à la nature vaste et décentralisée des WSN à laquelle s'ajoute la possible mobilité des nœuds participant au routage. Cacher le chemin revient à compliquer et alourdir le travail de ces protocoles déjà complexes tel que RPL.

Il existe cependant d'autres solutions d'anonymat. En effet, il est possible, plutôt que de tenter de cacher le chemin, d'appliquer les techniques d'anonymat aux adresses directement.

Park, Bang et al. décrivent dans [102] une méthode pour cacher les adresses destinations contenues dans l'en-tête MAC. La sécurité couche MAC est activée. Cette méthode utilise les filtres de Bloom. Cette structure probabiliste permet de vérifier de façon rapide la présence d'un élément dans un ensemble de données et ainsi de stocker plus d'information avec peu de mémoire. En effet, seul le résultat de fonctions de hachage est stocké. La mise en place de l'anonymat des adresses destinations se déroule en deux phases.

Lors d'une phase préliminaire, la station de base collecte les adresses MAC (a_i) des i nœuds du réseau à déployer. Au lieu de stocker les différentes adresses telle quelle, celle-ci va utiliser les filtres de Bloom (BF) pour stocker les résultats des fonctions de hachage pour chaque adresse. Cette station génère également des données aléatoires R_j ($j \neq i$) qu'elle stocke également dans son filtre de Bloom (BF(R_j)). Le filtre contient donc les informations concernant les nœuds mais également des informations aléatoires choisies pour ne pas chevaucher les adresses MAC et créer de faux positifs dans le filtre. Le réseau est ensuite déployé. Chaque nœud va calculer l'empreinte de son adresse MAC par le filtre de Bloom appelée bfid = BF(a_i). Lors de la phase de communication du réseau, lorsque la station de base veut communiquer avec un nœud, elle va remplacer l'adresse destination MAC du nœud par $recvID$. $recvID$ est calculé grâce au filtre de Bloom. Prenons l'exemple d'une communication avec le nœud $i = 3$. La station de base va choisir q (prenons $q = 4$) valeurs parmi les R_j valeurs aléatoires générées et stockées et calculer $recvID = BF(a_3, R_2, R_6, R_7, R_9)$. Lorsque le nœud reçoit la trame avec $recvID$, il va alors vérifier si son bfid est contenu dans $recvID$. Si c'est le cas, la trame lui est destinée. Il peut la traiter. L'inconvénient de cette méthode est qu'elle n'autorise que les communications de la station de base vers les nœuds et dans une topologie étoile. Elle limite également le nombre de nœuds dans le réseau. En effet, la station de base doit définir au préalable le nombre de nœuds qu'elle autorise afin de configurer son filtre de Bloom.

Toutes les méthodes d'anonymat citées précédemment compliquent donc le déploiement et le maintien des WSN et ne sont pour la plupart pas applicables dans les WSN à cause des contraintes qu'apportent ce genre de réseau. De plus, dès lors où l'on parle d'anonymat, le problème du contrôle d'accès se pose. En effet, l'identifiant est souvent utilisé pour gérer l'accès au réseau. Or, que ce soit dans l'anonymat du chemin ou de l'adresse en elle-même, ce contrôle est compliqué voire impossible. L'un des critères pour l'adoption d'une solution de protection de la vie privée est que celle-ci ne nuise pas à la sécurité. Empêcher le contrôle d'accès revient à négliger un mécanisme de sécurité.

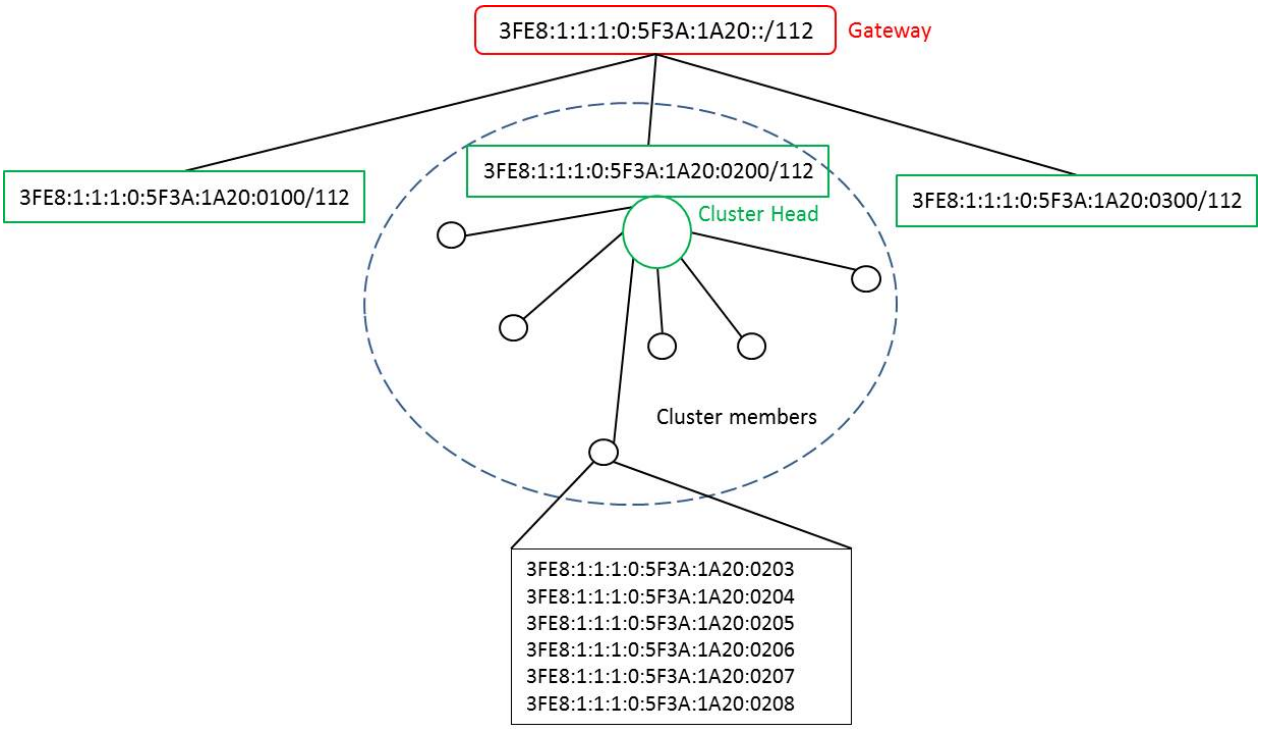
Une nouvelle approche consiste à ne pas utiliser l'anonymat pour une adresse mais à fournir des pseudonymes reliés aux identités afin de permettre au nœud d'utiliser ces derniers dans le réseau à la place des identifiants uniques mais également permettre le contrôle d'accès et donc l'authentification des émetteurs.

5.3 L'utilisation de pseudonymes

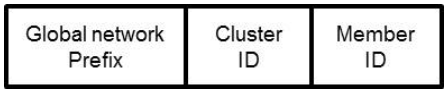
L'idée de l'utilisation de pseudonymes est de remplacer des identifiants long termes par des courts termes que l'on rafraichit. Pour cela, il est possible de permettre au nœud de choisir un identifiant parmi plusieurs. C'est l'utilisation des listes de pseudonymes.

5.3.1 Les listes de pseudonymes

Une première possibilité est donc de fournir aux nœuds une liste statique de pseudonymes prédéfinis, liste distribuée par une autorité de confiance. Ces méthodes s'appellent les Simple Anonymity Scheme (SAS). Wang et al. proposent dans [103] un schéma d'adressage basé sur les techniques SAS. Dans cette méthode, les nœuds sont organisés en *cluster* et la topologie est en étoile (cf. Figure 5.3(a)). Trois acteurs interviennent dans le réseau. La passerelle, qui assure le rôle d'autorité de confiance et initie le réseau. Elle va permettre dans un premier temps de diffuser le préfixe 3FE8 :1 :1 :1 :0 :5F3A :1A20 commun à toutes les adresses du réseau et utilisé pour former la première partie de celles-ci suivant le format de la Figure 5.3(b). Cette passerelle dispose également d'un ensemble d'adresses prédéfinies qui seront utilisées pour les communications dans le réseau. Chaque *cluster* du réseau est relié à la passerelle par un Cluster Head (CH) assurant le rôle de nœud puits. Afin d'organiser l'adressage, chaque CH va demander à la passerelle de lui attribuer une partie des adresses qu'elle possède. Pour cela, celle-ci scinde l'ensemble de départ en sous-ensembles. Ainsi, tous les membres d'un même *cluster* posséderont une partie cluster ID identique (Figure 5.3(b)). Prenons l'exemple du *cluster* du milieu de la Figure 5.3(a), la passerelle fournie au CH l'ensemble des adresses dont le "cluster ID" est 02. Le CH va alors disposer à son tour d'un sous-ensemble des adresses mises à disposition dans le WSN par la passerelle qu'il pourra fournir à ses membres. On peut parler de sous-réseau où le CH agit en tant qu'autorité de confiance pour les membres du *cluster*.



(a) Schéma d'adressage de [103].



(b) Format d'une adresse IPv6.

FIGURE 5.3 – Attribution d'une liste d'adresse.

Imaginons maintenant un nouveau nœud souhaitant rejoindre le réseau via le *cluster* dont l'ID est 02. Dans ce *cluster*, 5 nœuds forment déjà le réseau. Ce sont les Cluster Member (CM). Chacun de ces nœuds possèdent déjà une liste de pseudonymes à utiliser pour leurs communications avec le CH. Afin de réduire les informations disponibles pour un attaquant, les listes mises à disposition de chaque nœud ont toutes des tailles aléatoires. Le nouveau nœud va choisir le *cluster* auquel il souhaite appartenir et contacter le CH afin d'obtenir une liste d'adresses qu'il pourra utiliser pour communiquer au sein du réseau. Il doit également déterminer la taille de la liste de ses pseudonymes. Pour ce faire, il va tirer n, nombre aléatoire indiquant la taille de la liste d'adresses qu'il souhaite recevoir et envoyer une demande au CH contenant n. Le CH regarde s'il possède n adresses consécutives encore disponibles dans l'ensemble que lui a fourni la passerelle. Si ce n'est pas le cas, le CH calcule aléatoirement une nouvelle valeur pour n et recommence la recherche. Dès lors où un ensemble de n adresses consécutives est disponible, le CM se voit attribué cet ensemble et peut choisir aléatoirement une adresse pour communiquer et en changer périodiquement. Prenons ici $n = 10$, nombre obtenu par tirage aléatoire par le nouveau nœud. Dans la Figure 5.3(a), on peut voir que le nouveau nœud n'a pas obtenu 10 adresses mais uniquement 6 adresses allant de 3FE8 :1 :1 :1 :0 :5F3A :1A20 :0203 à 3FE8 :1 :1 :1 :0 :5F3A :1A20 :0208. Le CH n'avait donc pas assez d'adresses consécutives disponibles pour fournir les 10 demandées et a dû retirer un nouveau nombre aléatoire. Le nouveau CM possède donc la liste d'adresses dont le champ "Member ID" va de 03 à 08 ce qui permet de l'identifier de manière unique auprès du CH. En effet, afin d'authentifier les pseudonymes utilisés, le CH va maintenir une table permettant de lier l'adresse IPv6 réelle du CM et l'ensemble des adresses qui lui sont attribuées.

La méthode présentée s'apparente à du DHCPv6 où l'on met à disposition une liste d'adresses statiques réutilisables plutôt qu'une adresse unique.

L'avantage de cette solution est qu'il n'est pas obligatoire de procéder à du DAD. L'unicité des adresses est assurée par le choix de la liste de départ fournie à la passerelle puis scindée aux différents CH et CM. De plus, le contrôle d'accès est assuré par la table stockée par le CH.

En revanche, cette table utilise une grande place mémoire, ce qui oblige à avoir des CH de type FFD. De même, le CM doit également stocker tous ses pseudonymes ce qui peut être compliqué pour des nœuds contraints en mémoire. La topologie en étoile ainsi que la segmentation en cluster réduit les possibilités qu'offrent au départ les réseaux 6LoWPAN.

De plus, l'ensemble d'adresses fourni au CM est limité et les adresses réutilisables, si bien que lorsque la périodicité entre deux changements d'adresses est faible, les adresses peuvent revenir souvent. De même, un nœud ayant rejoint le réseau lors de l'établissement de celui-ci aura plus de chance d'obtenir un ensemble plus grand d'adresses disponibles (choix du n aléatoire possible plus grand) qu'un nœud rejoignant un réseau déjà dense. Elle ne règle donc pas le problème de protection de la vie privée si le nœud est statique. En effet, une observation du réseau permet à un attaquant d'identifier toutes les adresses. Il peut lier plusieurs pseudonymes ensemble. Les adresses pouvant être réutilisées, un attaquant peut utiliser cette propriété pour mener ses attaques ciblées comme dans le cas d'un réseau sans solution de protection de la vie privée. En revanche, lorsque le nœud va se déplacer, il pourra obtenir un nouvel ensemble et on ne pourra pas le lier avec les identités qu'il avait dans le précédent réseau.

Afin de pallier aux problèmes d'une liste statique dont les éléments sont réutilisables, une solution est d'utiliser une liste de pseudonymes temporaires. Les pseudonymes ayant une durée de vie, un nœud ne pourra pas réutiliser une même valeur de pseudonyme pour de futures communications.

Oualha, Olivereau et al. proposent dans [104] une variante de la méthode précédente où les éléments de la liste sont supprimés après utilisation. Dans celle-ci, un serveur d'authentification calcule tout d'abord, pour chaque nœud, une liste de pseudonymes à utiliser à la place des IID des adresses IPv6. Cette liste est ensuite transmise au nœud concerné. Pour chaque nœud, le serveur crée un filtre de Bloom associé contenant les pseudonymes générés. Les différents filtres de Bloom obtenus sont ensuite distribués de manière sécurisée aux différentes *gateway* du réseau ou autres points de contrôle. Ainsi chacun peut valider les messages reçus des nœuds. L'utilisation des filtres de Bloom permet de réduire la mémoire nécessaire à la mise en place du mécanisme de contrôle d'accès. Pour communiquer, un nœud va choisir dans la liste un pseudonyme, créer son adresse IPv6 suivant le protocole SLAAC mais avec le pseudonyme comme IID au lieu de l'adresse MAC. Il va ensuite envoyer sa trame avec ce pseudonyme comme adresse source réseau. La *gateway* va vérifier la présence de ce pseudonyme dans un des filtres de Bloom avant de la router ou de la supprimer si elle n'arrive pas à retrouver le pseudonyme. Les pseudonymes ne sont utilisables qu'une seule fois. Pour s'assurer de cette unicité, après routage, la *gateway* supprime du filtre de Bloom la présence de ce pseudonyme. Dès que tous les pseudonymes sont utilisés, il est nécessaire de recommencer toute la procédure de génération et de partage des différents éléments.

Grâce à cette solution, le problème de la réutilisation des adresses est réglé. La liste évolue dynamiquement grâce à la suppression des pseudonymes déjà utilisés. Les nœuds utilisent un protocole similaire à SLAAC afin de générer leurs adresses IPv6. Les nœuds de contrôle sont capables de procéder au contrôle d'accès grâce aux filtres de Bloom tout en réduisant la mémoire nécessaire au stockage des pseudonymes dans ces nœuds. Néanmoins, cela oblige le serveur d'authentification à prédéfinir la liste pour programmer les filtres de Bloom. Ce schéma n'est pas adapté à des nœuds trop contraints et pouvant se mettre en veille pour économiser de l'énergie. Lors de déploiement de WSN, les nœuds peuvent être éloignés les uns des autres et les communications ne sont pas fiables. Devoir compter sur une autorité de confiance pour obtenir les pseudonymes est donc compliqué dans ce genre d'environnement.

La solution précédente a néanmoins pour avantage de montrer l'intérêt de l'utilisation d'une liste de pseudonymes temporaires. Il est donc nécessaire de trouver une solution qui permette aux nœuds de générer leurs pseudonymes sans compter sur une autorité de confiance.

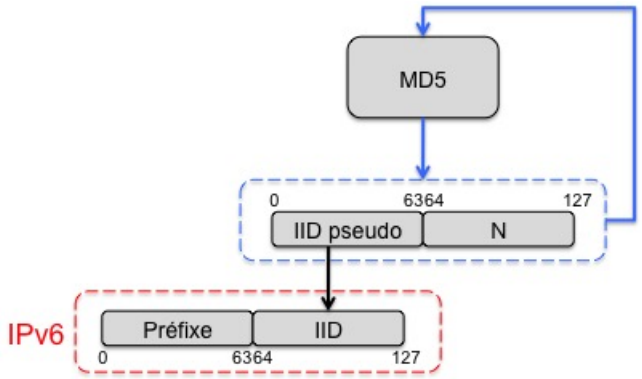


FIGURE 5.4 – Format d'une adresse IPv6 temporaire créée avec la RFC 4941.

C'est ce que proposent Narten, Draves et al. dans la RFC 4941 [105]. L'IETF est parti d'un constat similaire à celui fait dans ce mémoire sur les fuites d'identifiants existantes lors de l'auto configuration SLAAC grâce aux adresses MAC. Ils ont donc publiés la RFC "privacy extensions for stateless address autoconfiguration in IPv6" permettant de pallier aux problèmes de vie privée.

Les auteurs proposent l'utilisation d'adresses IPv6 temporaires dont la partie IID est générée à partir d'une fonction de hachage. La génération du pseudonyme est la suivante.
La configuration de la partie préfixe est conservée comme définie par SLAAC. Afin de générer les pseudonymes pour l'IID, les nœuds vont utiliser la fonction de hachage MD5 [106] (cf. Figure 5.4). La sortie de cette fonction a une taille de 128 bits. Or, la partie IID n'a besoin que de 64 bits. Le nœud ne va donc conserver que les 64 premiers bits de la sortie de la fonction de hachage comme prochain IID. Les 64 derniers bits (appelés N sur la Figure 5.4) seront stockés et concaténés avec la valeur de l'IID précédent pour ensuite être utilisés comme entrée de MD5 lors de la génération des futurs pseudonymes. Lors de la première génération de pseudonyme, la valeur de N est choisie aléatoirement. Dès lors où la durée de vie du pseudonyme est écoulée, de nouveaux pseudonymes sont générés.

Chaque nœud est ainsi capable d'auto générer une liste de pseudonymes. Le nœud va alors choisir pour chacune de ses communications sortantes un pseudonyme de la liste. C'est à dire que le nœud émetteur va choisir un pseudonyme différent pour chacune des communications qu'il possède dans le réseau avec les autres nœuds. La solution a été pensée pour les réseaux IP classiques. Les communications sortantes sont vues ici vis-à-vis des applications internet classiques. Ces connexions peuvent correspondre à du SSH ou encore du téléchargement de films.

Le renouvellement de la liste de pseudonymes est périodique. Afin de ne pas casser ces connexions et permettre le bon déroulement de l'application, le changement de pseudonyme doit respecter certaines règles. En effet, lorsque la durée de vie d'une adresse est atteinte, celle-ci doit être remplacée. Or, si celle-ci est utilisée dans une connexion, le changement peut nuire au déroulement de l'application comme par exemple le téléchargement. La RFC propose de rendre l'adresse inutilisable pour de nouvelles connexions mais de la garder pour celle déjà établie. A la fin de l'application l'adresse est supprimée et remplacée par un nouveau pseudonyme temporaire. La liste des pseudonymes disponibles évolue dynamiquement.

Cette méthode a pour avantage de fonctionner avec SLAAC. Les nœuds sont capables de générer des pseudonymes sans l'intervention d'une autorité de confiance. La connaissance des anciennes adresses ne permet pas de deviner la prochaine adresse. De même, la connaissance de l'adresse actuelle ne permet pas de retrouver et lier les adresses précédentes. Un pseudonyme ne sera utilisé que pendant sa durée de vie et pour un seul lien. Dès lors où la communication n'est plus établie le pseudonyme n'est plus utilisé. Un attaquant ne peut donc pas compter sur sa réutilisation pour mener ses attaques contrairement aux listes fixes.

En revanche, cette RFC a été pensée dans le but de pallier aux problèmes de vie privée et elle ne tient pas compte de la sécurité et ne définit aucun protocole sécurisé pour fonctionner avec. Elle est donc sujette à beaucoup d'attaques et notamment les attaques par usurpation d'adresses.

Cette méthode a pour inconvénient de ne chiffrer que l'adresse source et donc de laisser l'adresse destination en clair. Elle augmente la complexité de la mise en place de mécanismes de gestion du réseau. Le contrôle d'accès n'est pas possible car le nœud destinataire n'a pas les moyens de lier le pseudonyme et l'adresse de départ.

Pour conclure, les problèmes dus à l'utilisation d'une liste statique dans les SAS ont été réglés grâce à l'introduction d'une liste de pseudonymes temporaires fournie par une autorité de confiance. Néanmoins, l'intervention de cette dernière dans la génération des pseudonymes rend le déploiement dans les WSN très compliqués. Une solution a donc été publiée par la RFC 4941 afin de permettre aux nœuds d'auto générer leurs pseudonymes. Néanmoins, cette solution n'offre pas la possibilité d'authentifier les adresses et de faire du contrôle d'accès.

Il est donc important de mettre en place une solution de protection qui permette aux nœuds de générer leurs propres pseudonymes dynamiques et aux destinataires de pouvoir vérifier et lier le pseudonyme à une adresse réelle.

5.3.2 Les pseudonymes dynamiques

La première solution n'a pas été créée dans l'objectif d'assurer la protection des identifiants permanents. Elle découle d'une solution de sécurité. En effet, dans un réseau classique, la découverte de voisins se fait via le protocole NDP. Or, ce protocole n'étant pas sécurisé, il était sujet à des attaques. Afin de pallier à ce problème, l'IETF a introduit le protocole SEcure Neighbor Discovery (SEND) [107]. Dans cette version sécurisée du protocole de découverte des voisins, des options sont ajoutées afin de permettre de contrer les attaques par rejeu mais également permettre l'authentification des messages.

L'une des options disponibles et définies dans cette RFC s'appelle les Cryptographically Generated Addresses (CGA) [108]. L'utilisation des CGA et du protocole SEND permet de prouver la possession de l'adresse et donc de réaliser du contrôle d'accès tout en protégeant les identifiants. Pour cela, l'IID de l'adresse IPv6 est généré de manière cryptographique à partir d'un nombre aléatoire et de la fonction de hachage Secure Hash Algorithm 1 (SHA-1). Le nœud a besoin de posséder au préalable une paire de clés qui sera utilisée pour la génération et la vérification des pseudonymes.

L'algorithme de génération est le suivant :

1. Le nœud génère un nombre aléatoire appelé "modifier" de 128 bits.
2. Il concatène la valeur obtenue avec 9 octets nuls, la valeur de sa clé publique et des champs d'extensions si présents. La fonction de hachage SHA-1 est alors appliquée à la valeur obtenue. Les 112 premiers bits de la sortie sont gardés et appelés $Hash2$.
3. La sécurité des CGA repose sur un paramètre appelé SEC. Il détermine la résistance de l'adresse aux attaques par force brute. SEC indique le nombre de zéro devant se trouver au début de $Hash2$. Plus la valeur de SEC est élevée, plus le nombre de zéros est important. Dans cette étape, le nœud compare les 16*SEC premiers bits de $Hash2$ avec zéros. Si le nombre est insuffisant, la valeur de "modifier" est incrémentée et l'étape 2 est recommencée. En revanche, si le nombre correspond, alors l'algorithme passe à l'étape 4.
4. Un compteur de collisions de 8 bits est utilisé pour déterminer l'unicité de l'adresse. Cette étape initialise à 0 cette valeur.
5. La valeur finale obtenue pour le "modifier" est concaténée avec la valeur du préfixe réseau, le compteur de collisions, la clé publique et des champs d'extension si présents. La fonction de hachage est alors appliquée à la valeur obtenue. Seuls les 64 premiers bits sont conservés pour former $Hash1$.
6. L'IID est ensuite créé à partir de $Hash1$ dans lequel les trois bits les plus à gauche sont modifiés pour indiquer la valeur de SEC et où les bits 6 et 7 sont mis à 0. Cette dernière modification est réalisée afin d'obtenir une adresse respectant le format EUI-64.
7. L'adresse IPv6 est alors formée de manière classique grâce au préfixe réseau et à cet IID créé.
8. Afin de s'assurer de l'unicité de l'adresse, le protocole DAD est ensuite lancé. Si une collision est détectée, le compteur de collision est alors incrémenté de 1 et l'algorithme recommence à l'étape 5.

Cette détection peut avoir lieu au maximum trois fois. Au delà, la procédure est avortée et aucune adresse CGA n'est créée.

9. Enfin, une structure est créée contenant la valeur finale du "modifier" concaténée avec le préfixe, le compteur de collisions final, la clé publique et les champs d'extension. Cette structure est appelée "CGA Parameters".

A la fin de la procédure, le nœud possède donc un pseudonyme et une structure. Il va donc pouvoir remplacer son adresse IPv6 par ce CGA. Afin que le receveur de la trame puisse vérifier que l'adresse est légitime et est bien celle associée à la clé publique du nœud source, il a besoin de connaitre deux éléments. Il doit d'abord connaitre l'adresse CGA mais également la structure "CGA Parameters". Le nœud peut les partager grâce au protocole SEND qui permet l'ajout de ces nouveaux champs dans la trame. Les informations nécessaires à la génération de l'adresse sont donc envoyées dans la trame.

La vérification se déroule de la manière suivante :

1. Le nœud vérifie que le compteur de collisions ne possède pas une valeur supérieure à 2. Dans le cas contraire la vérification avorte.
2. Il vérifie ensuite que le préfixe de l'adresse IPv6 correspond au préfixe utilisé pour la génération du pseudonyme et contenu dans la structure. Dans le cas contraire la vérification avorte.
3. Il exécute la fonction de hachage SHA-1 sur la structure et garde les 64 premiers bits de la sortie afin d'obtenir la valeur $Hash1$.
4. Il compare alors cette valeur de $Hash1$ avec l'IID. Les différences dues à l'insertion de la valeur de SEC dans l'IID ainsi que la mise en conformité avec les adresses EUI-64 sont ignorées. Si des différences apparaissent sur les autres bits, la vérification échoue.
5. Le nœud récupère ensuite la valeur de SEC des trois premiers bits de l'IID.
6. Le nœud concatène la valeur du "modifier" avec 9 octets nuls, la clé publique et les champs d'extensions. La fonction de hachage est appliquée à la valeur obtenue et les 112 premiers bits de la sortie sont récupérés pour former $Hash2$.
7. Enfin, le destinataire vérifie que les 16*SEC premiers bits de $Hash2$ sont bien tous nuls. Si ce n'est pas le cas, la vérification échoue.

Les nœuds peuvent également signer leurs messages via l'utilisation de leur clé privée. Le fonctionnement des CGA fait qu'il n'est pas nécessaire de compter sur une autorité pour vérifier une adresse. Elles sont auto certifiées.

L'un des inconvénients des CGA est qu'il n'est pas adapté aux adresses Unicast ni à l'utilisation de proxy. Afin d'étendre l'utilisation des CGA aux cas précédents, Cheneau et al. dans [109] ont défini un nouveau concept permettant à une adresse de posséder plusieurs clés publiques lui étant associées.

Comme dans la RFC 4941, la solution CGA ne permet de cacher que l'IID de l'adresse source mais pas celui de l'adresse destination. De plus, comme l'expliquent Rafiee et al. dans [110] la valeur de SEC est généralement choisie nulle ou égale à 1 car au delà la vérification pourrai prendre des heures voire des années avec des CPU modernes non contraints. C'est notamment cette étape permettant de s'assurer de la résistance de l'adresse face à des attaques par force brutes qui coûte le plus cher dans le processus de génération des adresses. La génération des CGA consomme donc considérablement de l'énergie et il est difficile de répéter le processus souvent sur des nœuds contraints sous peine d'épuiser les ressources. Ce processus est trop gourmand pour un déploiement en milieu contraint. Il va donc être utilisé lors de la phase d'association et ne sera jamais mis à jour, ce qui revient à avoir un IID statique. En conclusion, les CGA sont intéressantes pour le monde de l'IP classique car elles permettent de fonctionner avec SLAAC et SEND et autorisent le contrôle d'accès mais sont trop coûteuses en création pour les WSN et ne protègent que l'adresse source.

Rafiee et al. dans [110] ont définis un nouveau schéma d'adressage appelé Simple Secure Addressing Scheme (SSAS) et inspiré du fonctionnement des CGA. Le but de ce nouvel algorithme est d'améliorer les performances des CGA tout en fonctionnant avec SEND.

Dans cette solution, un nœud va utiliser les ECC. Les courbes elliptiques offrent de nombreux avantages par rapport à RSA en terme d'implémentation [111]. L'algorithme de génération de l'IID est le suivant :

1. Le nœud doit générer une paire de clés de 192 bits grâce à ECC.
2. La clé publique est coupée en deux parties de 96 bits chacune.
3. Dans chacune des parties obtenues à l'étape 2, un IID partiel est prélevé en prenant les 32 premiers bits de chaque partie.
4. L'IID final est alors configuré en concaténant les deux IID partiels.

Les adresses IPv6 sont alors configurées avec SLAAC en utilisant ce nouvel IID. Pour la configuration de l'adresse IPv6 globale, le nœud doit procéder à du DAD afin de s'assurer de l'unicité de l'adresse.

Tout comme pour les CGA, SSAS offre la possibilité au nœud de signer son message grâce à la clé privée. Les messages sont signés avec la clé privée du nœud. Cette signature permet de protéger les nœuds des attaques *spoofing* et de fournir une preuve de possession de l'adresse générée.

Le nœud peut maintenant utiliser ces nouvelles adresses pour communiquer dans le réseau. Afin de créer un climat de confiance, SSAS introduit une autorité capable de lier une adresse MAC et une clé publique et de fournir, si demandée, une preuve de légitimité d'un nœud aux autres nœuds du réseau.

Tout comme avec les CGA, le nœud destinataire est capable de vérifier l'adresse reçue. Pour cela, le nœud source doit également ajouter des champs supplémentaires à la trame transmise et notamment pour la signature et sa vérification. Il utilise également SEND pour établir une découverte de voisins sécurisée. L'algorithme de vérification de l'adresse est le suivant :

1. Le nœud récupère le *timestamp* qui a été ajouté aux messages par le nœud émetteur et l'appelle $t1$.
2. Le nœud va également récupérer un *timestamp* local via son propre système. Ce *timestamp* s'appelle $t2$.
3. Le nœud vérifie alors que le message a bien été reçu entre $t2 - x$ et $t2 + x$. La valeur de x dépend du réseau, des retards admissibles dus aux caractéristiques de ce dernier... Si la valeur de $t1$ n'est pas dans cet intervalle, le message est supprimé. Sinon le nœud passe à l'étape 4. Cette étape permet d'éviter les attaques par rejeu.
4. Le nœud récupère ensuite la clé publique de l'émetteur.
5. Le nœud va alors effectuer les mêmes étapes que le nœud émetteur sur la clé publique. Il va ensuite comparer la valeur obtenue avec la valeur contenue dans la trame. Si des différences apparaissent, la vérification échoue.

Rafiee et al. proposent dans [110] de comparer les performances de leur solution avec une implémentation des CGA. Leurs tests sont menés dans des environnements non contraints. D'après leurs résultats, le temps nécessaire pour la génération de la paire de clés puis d'une adresse CGA est d'environ 1000 milli secondes contre moins de 75 milli secondes pour les SSAS. SSAS est donc plus rapide en calcul qu'avec CGA. Fort de ces résultats, les auteurs affirment que malgré que leur solution soit prévue pour les réseaux classiques comme CGA elle peut néanmoins être utilisée dans les WSN.

Cette solution permet donc d'utiliser des pseudonymes dynamiques avec un coût de génération plus faible que les CGA tout en réutilisant les mécanismes de sécurité proposés dans le protocole SEND. Il protège également contre certaines attaques de *spoofing* et permet le contrôle d'accès. Il est également compatible avec SLAAC. Néanmoins, ce schéma repose sur une autorité de confiance. Or, nous avons expliqué précédemment que lors de déploiement de WSN les contraintes rendaient l'utilisation d'une autorité de confiance compliquée. De plus, cette méthode nécessite également de procéder au protocole DAD très coûteux. Si le changement d'adresses apparait trop souvent, cela apporte un surcoût de communications pour DAD.

Cette solution comme les précédentes a été pensée pour un déploiement dans des réseaux IP classiques. C'est pourquoi, elle se base sur des protocoles comme DAD ou encore des calculs qui sont trop coûteux pour être réalisés dans des environnements contraints. Il est alors important de mettre en place une solution pensée en concordance avec l'environnement atypique des WSN et les problèmes qu'il apporte.

Afin de générer un pseudonyme à partir d'une graine autre qu'un nombre aléatoire qui ne permet pas le contrôle d'accès, ou d'une clé publique qui nécessite un surcoût de trame pour assurer l'unicité, Tunaru,

Denis et al. dans l'article [112] préconisent d'utiliser, pour les réseaux ad-hoc, des informations de la couche Physique. Par exemple, il est possible d'utiliser le RSSI, la position ou la distance relative ainsi que les informations de connectivité pour obtenir, de manière cryptographique par fonction de hachage, un pseudonyme. Leur technique permet d'éviter les attaques par usurpation d'identité.

La propagation des pseudonymes peut se faire de deux manières : soit par communication directe en sécurisant cette dernière, soit par inférence. Les nœuds vont pouvoir, dans ce cas, deviner les pseudonymes de leurs voisins. Pour cela, il est nécessaire que les métriques utilisées pour la génération du pseudonyme soient réciproques et stables. C'est cette génération grâce au RSSI et au Round Trip - Time of Flight (RT-ToF) qui est préconisée par [112]. Cette méthode a pour avantage de ne pas introduire de complexité supplémentaire car il n'y a pas de surcoût dans les échanges de trames ni de calculs supplémentaires sur les données mesurées. Tunaru, Denis et al. dans [112] ne précisent pas si le changement de pseudonymes arrive fréquemment et si celui-ci influe sur les performances du réseau. Aucune information n'est donnée sur l'avertissement envoyé aux voisins concernant le changement de pseudonymes ainsi que sur le fait que l'ancien n'est plus valable et qu'il ne faut plus l'utiliser. De plus, le routage n'est pas stipulé. On peut penser que, comme on est dans les réseaux ad-hoc, le routage est semblable au routage AODV qui diffère quelque peu dans son fonctionnement d'avec RPL et est énergivore. En revanche, elle montre l'intérêt qu'il y a à permettre au nœud de calculer de façon réciproque son pseudonyme mais également ceux de ses voisins, tout en fonctionnant avec les protocoles mis en place (RPL, SLAAC).

Afin de limiter l'intervention d'autorité de confiance mais également pour rendre l'auto configuration plus rapide et plus flexible, il est nécessaire de laisser les nœuds auto générer leurs pseudonymes de la même façon qu'ils généraient leurs adresses IPv6 avec la méthode SLAAC.

La RFC 7217 [113] propose une méthode de génération d'adresses IPv6 basée sur SLAAC mais n'utilisant pas l'adresse MAC. Pour cela, le nouvel IID est calculé grâce à une fonction cryptographique telle que SHA-1. Elle prend en entrée le préfixe IPv6 du réseau, des identifiants spécifiques au réseau (par exemple, SSID) et à l'implémentation (adresse source). Elle prend également en compte un compteur pour la résolution des conflits avec le protocole DAD et une clé secrète. Tous les bits de l'IID sont traités de manière opaque. Cela implique qu'il n'y a pas, comme c'est le cas pour les adresses IPv6 classiques, de bits réservés pour indiquer le caractère local ou universel de l'adresse. Le préfixe entrant en compte lors du calcul de l'IID, ce dernier est alors différent pour chaque adresse IPv6 utilisée (locale, globale). Cette solution empêche les attaques par scan d'adresse ou encore par exploitation des caractéristiques matérielles. Elle permet également d'empêcher le suivi d'un nœud lorsque celui-ci se déplace entre deux réseaux. En revanche, un nœud gardant la même connexion au sein d'un réseau peut être suivi tant que celle-ci est valide. Dans les réseaux de capteurs, les nœuds sont généralement en mouvement à l'intérieur du réseau mais appartiennent de manière prolongée au même réseau. Cette solution n'est donc pas optimale pour ce type de réseau. De plus, cette solution ne permet l'utilisation de pseudonymes que pour les adresses IPv6. Les adresses MAC qui sont aussi des identifiants permanents restent sensibles à de l'écoute passive et il est donc toujours possible de mener de attaques ciblées en utilisant cette adresse.

Groat et al. proposent dans [114] une solution appelée Moving Target IPv6 Defense (MT6D). Elle permet aux nœuds de générer les adresses source et destination IPv6 des trames. Les différents nœuds vont partager une clé symétrique. La distribution des clés n'est pas spécifiée dans l'article [114]. Cette solution, contrairement aux précédentes permet l'utilisation de pseudonymes pour les adresses IPv6 mais également pour les adresses MAC. Le schéma MT6D fournit une protection contre les intrusions, permet la dissimulation des identifiants permanents et ne nécessite pas de se ré authentifier à chaque changement d'adresse. MT6D peut être vu comme du saut de canal mais appliqué aux adresses. MT6D nous a paru être la solution la mieux adaptée aux contraintes des réseaux de capteurs et permettant de protéger simultanément adresses MAC et IPv6. Le *design* de MT6D permet également le contrôle d'accès et l'authentification.

5.4 Conclusion

Lorsque l'on souhaite mettre en place une solution de protection des identifiants dans les WSN, celle-ci doit permettre de dissimuler l'identité du nœud tout en respectant les protocoles en place (RPL et SLAAC).

Elle doit également cohabiter avec la sécurité, notamment en terme de contrôle d'accès. Enfin, le système doit conserver des performances acceptables.

Les solutions détaillées dans cet état de l'art se classent en deux catégories : anonymat et utilisation de pseudonymes.

La Table 5.1 récapitule les avantages et inconvénients de chaque méthode et analyse la faisabilité sur des réseaux de capteurs. La première ligne indique si la solution est adaptable/adaptée (✓) aux WSN ou non (✗). Le contrôle d'accès étant un critère de sécurité important, la deuxième ligne indique la possibilité de faire cohabiter le schéma de dissimulation des identités et le contrôle d'accès. La ligne concernant la sécurité indique si les solutions ont été pensées en cohérence avec les mécanismes et protocoles de sécurité déjà existants.

Deux méthodes pour générer les pseudonymes ont été abordées. Le tableau rappelle si les pseudonymes sont fournis de manière statique (marquage S), c'est à dire que l'on fournit une liste fixe prédéfinie, ou dynamique (marquage D dans le tableau). Les techniques d'anonymat n'étant pas concernées par ce critère, NC pour Non Concerné est indiqué dans la case correspondante.

Le respect des protocoles utilisés pour les réseaux 6LoWPAN est également rappelé. Deux protocoles sont importants dans les WSN 6LoWPAN : le protocole de génération des adresses IPv6 SLAAC et le protocole de routage RPL.

DHCPv6 indique que les adresses n'utilisent pas un schéma similaire à SLAAC mais plutôt un adressage de type DHCPv6 fourni par un nœud jouant le rôle de serveur d'adresses. Dans le cas des solutions Privacy extension, CGA et SSAS, le protocole de découverte des voisins n'est pas RPL mais NDP ou sa version sécurisée SEND.

Enfin les performances et l'inconvénient majeur de chaque solution sont indiqués. Un signe − indique une perte de performance pour le réseau, un signe = indique qu'a priori la solution n'impacte pas ou peu les performances. Plus le nombre de signes − est élevé, plus les performances sont mauvaises par rapport à un réseau sans solution de dissimulation déployée. Les performances de chaque solution sont comparées.

Les méthodes d'anonymat étant lourdes, elles sont inadaptées pour les réseaux de capteurs. Les méthodes par listes sont intéressantes mais équivalent à du DHCPv6 ce qui est un inconvénient dans les WSN. Enfin, les méthodes qui semblent le mieux fonctionner sont celles où il est donné aux nœuds la possibilité de partager un secret leur permettant de calculer de façon réciproque les pseudonymes sans avoir à les propager ni à vérifier l'unicité. En revanche, étant prévue pour de l'IPv6 classique, les solutions proposées fonctionnent pour la plupart avec le protocole SEND et non avec RPL. Enfin, beaucoup des articles retenus pour l'état de l'art ne donnent que la méthodologie d'obtention des pseudonymes mais ne stipulent rien sur le rafraichissement ou l'avertissement de changement de pseudonymes, ni sur le type de réseau pour lequel la méthodologie a été pensée. Or, l'une des prérogatives de la protection de la vie privée est que la solution doit respecter les protocoles. Il est donc compliqué de vérifier ce critère sur ces solutions.

La solution la plus intéressante et respectant le plus les critères est celle de MT6D. MT6D permet de générer les adresses source et destination par fonction de hachage grâce à un secret commun. Cette solution sera présentée dans la partie suivante.

	TOR	[101] waypoint	[103] SAS	[104] Bloom filter	Privacy extension	CGA	SSAS	RFC 7217 [113]	MT6D
WSN	✗	✗	✓	✗	✗	✗	✓	✓	✓
Contrôle d'accès	✗	✗	✓	✓	✗	✓	✓	✓	✓
Sécurité	✓	✓	✓	✓	✗	✓	✓	✓	✓
Pseudonyme statique (S) / Dynamique (D)	NC	NC	S	D	D	D	D	D	D
Compatibilité avec SLAAC	✓	✓	DHCPv6	✓	✓	✓	✓	✓	✓
Respect de RPL	✗	✗	Topologie étoile	✗	NDP	SEND	SEND	✓	✓
Performance du réseau	- - -	- -	=	-	-	- -	-	=	-
Inconvénient majeur	Latence	Surcoût	Statique	Autorité de confiance	Contrôle d'accès	Gourmand	DAD	Adresses MAC en clair	Surcoût de trames de contrôle

Table 5.1 – Comparaison des différentes méthodes.

Partie 6

Solutions visant à préserver la confidentialité des adresses

Dans cette partie, MT6D, solution de l'état de l'art pour la dissimulation des adresses, est présentée. Une analyse théorique de son fonctionnement met en lumière certaines lacunes rendant le déploiement de celle-ci dans les réseaux de capteurs coûteux et non optimal. Notre vision du cahier des charges parfait d'une solution visant à préserver la confidentialité des adresses est donnée. Ephemeral est alors proposé pour corriger certaines des lacunes de MT6D. L'adéquation de notre solution avec le cahier des charges est étudiée.

6.1 Analyse de MT6D

Les premières solutions visant à préserver la confidentialité des adresses ont été créées pour le monde de l'IP classique. Lorsque le problème s'est posé pour les réseaux contraints, la première idée a consisté à adapter les solutions classiques à ce nouveau modèle.

Cette approche peut sembler intéressante car elle permet de récupérer des outils déjà utilisés et éprouvés afin de les adapter aux besoins des WSN. Néanmoins, les contraintes ainsi que la nature des réseaux ont très vite montrés les faiblesses d'une telle approche.

Des solutions dédiées ont donc commencées à voir le jour. Après un état de l'art, la solution MT6D nous a parue être la mieux adaptée à notre contexte.

Nous avons donc voulu analyser le comportement de MT6D.

6.1.1 Présentation de MT6D

Le schéma MT6D permet de cacher le trafic dans un réseau 6LoWPAN.

Deux contributions ont été proposées par Groat et al.

Premièrement, dans la version pensée pour les réseaux classiques, MT6D permet d'obfusquer la partie IID des adresses IPv6 source et destination grâce à l'utilisation de la fonction cryptographique SHA256. Le but de l'algorithme est de définir une fenêtre temporelle t_i pendant laquelle chaque nœud possède un pseudonyme IPv6 qu'il aura auto généré mais qui pourra être calculé également par les nœuds destinataires. Le pseudonyme est obtenu de la manière suivante :

$$IID'_{x(i)} = H[IID_x||K_s||t_i]_{0\rightarrow 63},$$

où, $||$ représente la concaténation, $IID'_{x(i)}$ représente l'IID caché pour le nœud x au temps t_i, IID_x représente la valeur initiale de l'IID pour le nœud x, K_s est la clé symétrique partagée par les nœuds du réseau et t_i le temps à l'instant i. $H[.]_{0\rightarrow 63}$ est la fonction cryptographique SHA256 dans laquelle seuls les 64 premiers bits de la sortie sont conservés pour générer l'IID obfusqué.

Une synchronisation est donc nécessaire pour s'assurer que tous les nœuds du réseaux changent leurs pseudonymes au même moment et possèdent la même valeur de t_i. Le protocole Network Time Protocol (NTP) est utilisé.

Lorsqu'un nœud rejoint le réseau, la première phase est réalisée de manière identique à un réseau n'utilisant pas de pseudonymes. Il va réaliser les phases de *join* et d'association dans lesquelles il va calculer son adresse IPv6, qu'il aura configurée grâce au mécanisme SLAAC et à son adresse MAC. Il va également récupérer les adresses de ses voisins et créer ses tables de routage et des voisins. Il va ensuite tenter de se synchroniser via le protocole NTP.

Dès que la synchronisation a réussi, il peut commencer le travail de calcul dynamique de pseudonymes temporaires. Il va alors calculer la valeur de son pseudonyme à utiliser pour son IID mais également pour toutes les adresses des nœuds destinataires. Ainsi, un nœud est capable de calculer les adresses IPv6 des destinataires finaux et donc de communiquer plus facilement. Il existe une réciprocité dans le calcul qui évite l'obligation de diffusion des nouvelles adresses. Seuls les routeurs locaux sont avertis du changement afin de permettre le routage correct de la trame. Néanmoins, la valeur de t_i doit être choisie de manière à permettre la réception de la trame avant la modification du pseudonyme. Elle est donc liée au temps d'aller d'un message (Single-Trip Time STT).

Ces nouveaux IID calculés, il va générer les adresses globales et locales correspondantes. Ces pseudonymes réalisés, les communications peuvent alors avoir lieu dans le réseau jusqu'à ce que la fenêtre temporelle t_i soit terminée. Si t_i n'est pas écoulée, les nœuds peuvent envoyer ou recevoir des trames de manière classique en remplaçant les champs d'adresses IPv6 par les pseudonymes calculés. Si la fenêtre est terminée, les nœuds doivent alors de nouveau calculer les pseudonymes qu'ils utiliseront pour leurs futures communications. Les anciennes adresses sont alors supprimées de manière à éviter les attaques par rejeu.

Les auteurs ont également démontré la faisabilité du changement dynamique d'adresses pour les réseaux de capteurs grâce à l'implémentation dans Contiki 2.7. Cette contribution permet d'élargir l'utilisation des pseudonymes aux adresses MAC. Dans l'article [115], Preiss, Sherburne et al. détaillent cette implémentation mais également les changements à effectuer dans les adresses IPv6 et MAC de Contiki. Contrairement à la fonction de hachage initiale, cette preuve de concept utilise des compteurs pour remplacer les valeurs des IID. Pour cela, le dernier octet de l'adresse IPv6 globale est incrémenté de 1 toutes les $t_i = 10s$. La nouvelle valeur de l'IID de l'adresse IPv6 est utilisée comme adresse MAC permettant ainsi l'utilisation de pseudonymes pour les deux adresses. Les auteurs donnent les mémoires/variables impactées et les fonctions à utiliser. Après changement d'adresses, les nœuds ne mémorisent plus les adresses précédentes. Celles-ci sont supprimées définitivement et remplacées par les pseudonymes.

Contrairement à la version classique, les nœuds n'utilisent pas la valeur de t_i dans le calcul des pseudonymes. La fenêtre est utilisée uniquement pour planifier les nouvelles générations de pseudonymes. Dans ce cas, le nœud ne sait calculer que ses pseudonymes MAC et IPv6. Les nœuds doivent alors procéder à l'avertissement de leurs voisins en leur envoyant leurs nouvelles adresses. Les nœuds procèdent donc à la reconstruction de l'arbre de routage pour s'assurer que tous possèdent les bonnes adresses. A chaque changement de pseudonymes, le réseau et l'arbre de routage se reconstruisent comme ils le feraient lors de la phase d'association. MT6D utilise les trames DIO et DAO de RPL pour transmettre les nouvelles informations sur le pseudonyme.

Prévu au départ pour le réseau classique IP, la thèse de Sherburne [116] adapte la méthode à un WSN composé d'un 6BR et d'un nœud. Néanmoins, les premiers résultats montrent que l'empreinte mémoire due à l'ajout de MT6D est importante.

6.1.2 Analyse théorique

MT6D permet de fournir des pseudonymes dynamiques pour les adresses MAC et IPv6. Grâce à cela, les attaques DoS basées sur l'adresse mais également les attaques MITM sont évitées. Elle permet également de préserver les identifiants.

Néanmoins, son fonctionnement possède des lacunes pour le déploiement dans les réseaux de capteurs. Tout d'abord, contrairement au fonctionnement classique d'adressage par SLAAC, l'adresse IP n'est pas

déduite de l'adresse MAC. C'est le pseudonyme utilisé pour l'adresse MAC qui découle de la modification de l'adresse IP. Ce protocole n'est défini par aucun standard.
L'utilisation d'une fonction de hachage est coûteuse en embarqué.

MT6D est sensible aux problèmes de désynchronisation des nœuds. En effet, tous les nœuds doivent changer leur pseudonyme au même moment. Dans un réseau de capteurs, plusieurs situations peuvent amener à une désynchronisation des nœuds.
Tout d'abord, les nœuds peuvent être ajoutés à des moments différents. Leurs horloges ne sont alors pas synchronisées et la période t_i qui définit le changement de pseudonyme ne sera pas synchronisée entre tous les nœuds.
De même, une désynchronisation de cette période peut apparaitre car les horloges internes de chaque nœud ne sont pas liées et elles peuvent dériver entre elles. Celles-ci n'étant plus synchronisées, les nœuds ne lancent pas au même moment la procédure de reconstruction du DODAG.

Le chronogramme de la Figure 6.1 montre l'effet d'une désynchronisation sur la reconstruction du DODAG. Lorsque les couleurs des slots sont identiques, cela indique que les nœuds ont des pseudonymes générés sur la même période t_i. Un changement de couleur indique un changement de pseudonyme pour le nœud.
Dans le chronogramme, trois nœuds fonctionnent avec MT6D. Lors des trois premiers changements de pseudonymes les horloges sont alignées ou possèdent une légère dérive. Lorsqu'un nœud lance alors la reconstruction de l'arbre de routage les autres nœuds ont déjà calculés la nouvelle valeur de leur pseudonyme et la reconstruction peut avoir lieu. Une seule reconstruction est nécessaire pour que chaque nœud avertisse la nouvelle valeur de pseudonyme.
En revanche, dans la deuxième phase, le nœud 1 lance une reconstruction alors que la période t_i des deux autres nœuds n'a pas encore été écoulée. Le nœud 1 va donc avertir son nouveau pseudonyme (couleur rouge) et reconstruire l'arbre avec celui-ci. Quant aux deux autres, la reconstruction se fera avec la valeur des pseudonymes précédentes (couleur orange). Lorsque le nœud 3 voit sa période t_i s'achever, il lance à son tour une reconstruction pour avertir de son nouveau pseudonyme (couleur rouge) sans tenir compte de la reconstruction précédente.

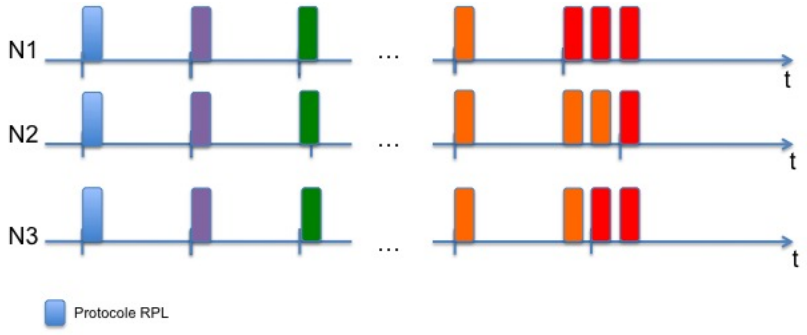


Figure 6.1 – Problème de désynchronisation de MT6D.

Ce comportement augmente alors le nombre de trames RPL nécessaires à la mise à jour des tables. Il participe à la détérioration des performances réseau car l'intervalle de temps nécessaire à la complète mise à jour des pseudonymes augmente avec le temps. Des techniques de resynchronisation des horloges sont alors nécessaires. Néanmoins, dans des réseaux vastes où la perte de paquets est autorisée, la mise en place de mécanismes tel que NTP est compliquée.

Enfin, à chaque changement de pseudonyme, les nœuds MT6D oublient leurs anciennes adresses et se comportent comme des nouveaux nœuds souhaitant rejoindre le réseau. Cette reconstruction complète du DODAG provoque un surcoût de trames de contrôle. Elle nécessite également de mettre à jour les tables de routage. Il est donc nécessaire d'estimer dans l'OS l'impact sur les tables de routage et le trafic.

6.2 Cahier des charges pour une solution idéale

L'analyse théorique de MT6D nous a permis d'identifier, ce qui pour nous, représentent les spécifications que doit avoir une solution d'utilisation de pseudonymes dynamiques idéale.

- La solution doit être flexible vis-à-vis des spécificités du réseau. Elle doit donc permettre l'utilisation des pseudonymes dans un réseau où des paquets sont perdus, des nœuds sont mobiles, peuvent être ajoutés ou disparaitre. Il faut que la solution ne compte sur aucune autorité pour la génération des pseudonymes afin de permettre le déploiement de vastes réseaux.
- L'introduction d'un mécanisme de sécurité ou de protection de la vie privée ne doit pas nuire aux protocoles déjà déployés. Plus particulièrement, le déploiement d'une solution visant les identifiants ne doit pas nuire aux protocoles utilisant ces identifiants. Il faut donc qu'elle soit compatible avec les protocoles d'adressage et de routage. La solution de dissimulation des identifiants doit reposer sur une norme ou des protocoles déjà existants. C'est pourquoi il est nécessaire que le fonctionnement de SLAAC, qui impose que l'adresse IP soit construite à partir de l'adresse MAC, soit gardé.
- Toujours dans ce soucis de compatibilité avec les protocoles déjà établis, les tables de routage et des voisins ne doivent pas être retouchées. L'évolution (ajout ou suppression d'entrées) doit suivre le comportement classique d'un réseau sans solution de dissimulation des adresses. Le protocole de maintien et l'association définis dans RPL doivent conserver leurs comportements.
- Le hachage étant trop coûteux en embarqué, une autre solution doit être adoptée pour la génération des pseudonymes.
- La diffusion et le partage des pseudonymes et/ou du matériel nécessaire pour la génération des pseudonymes ne doit pas perturber le fonctionnement de l'application. Dans la mesure du possible, il ne doit pas introduire de trafic supplémentaire ni de coût pour le réseau.
- Les nœuds doivent pouvoir être désynchronisés sans pertes de communications. Idéalement, l'utilisation des pseudonymes ne doit pas reposer sur une synchronisation des nœuds.

Ce cahier des charges permet de fournir une base de réflexion afin de construire la solution la plus adaptée aux réseaux de capteurs contraints.

6.3 Ephemeral

Pour pallier aux inconvénients de MT6D et respecter au mieux le cahier des charges établi précédemment, nous proposons Ephemeral, une solution de génération de pseudonymes pour les WSN multi sauts. Notre solution permet le masquage des adresses MAC et IPv6. Contrairement à MT6D, notre solution ne nécessite pas de reconstruire les tables de routage à chaque changement de pseudonymes, ni la synchronisation des nœuds.

6.3.1 Présentation

Le fonctionnement d'Ephemeral repose sur le déploiement de la sécurité couche MAC. Les nœuds utilisent les mécanismes de sécurité fournis par le standard IEEE 802.15.4 avec un niveau de sécurité au moins égal à 4. Il permet d'assurer la confidentialité, l'intégrité et l'authenticité du *payload* MAC. Pour cela, le chiffrement mais également l'authentification sont activés.

Tous les nœuds du réseau stockent deux clés secrètes partagées : une clé LK et une clé à court terme K_t qui peut être mise à jour et est utilisée pour dissimuler les adresses source et destination.

Le réseau 6LoWPAN est sécurisé grâce à la clé LK. Un nœud souhaitant rejoindre le WSN devra effectuer une association sécurisée durant laquelle la clé K_t sera chiffrée par LK et lui sera transmise. La mise à jour future de la clé K_t pourra être réalisée de la même manière. La génération ainsi que la distribution de la clé de sécurité LK ne sont pas considérées dans cette étude.

L'association se déroule de manière classique comme expliqué dans la partie 3 via le protocole défini par RPL. Aucune trame supplémentaire n'est introduite. Le *join* et l'association sont simplement réalisés avec la sécurité MAC activée. Cette solution permet de contrôler l'accès au réseau. En effet, seuls les nœuds possédant la clé LK pourront s'associer correctement.

Les adresses IPv6 sont chiffrées par la sécurité MAC et la clé LK. Seules les adresses MAC restent accessibles en clair par écoute passive. La Figure 6.2 montre le format simplifié d'un paquet échangé dans un réseau IEEE 802.15.4 sécurisé à la couche MAC.

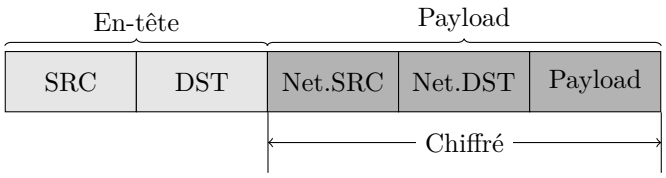


FIGURE 6.2 – Structure d'un paquet (simplifiée).

Ephemeral a donc pour but de remplacer les adresses encore transmises en clair et utiles au routage par des pseudonymes. Ces pseudonymes sont générés et vérifiés à l'aide d'un AES en mode compteur et de la clé secrète K_t sans intervention d'une autorité de confiance.

Deux planifications sont prévues avec Ephemeral :

- Périodique. Les pseudonymes sont mis à jour à intervalles de temps réguliers.
- Sur évènement. Les pseudonymes sont prévus pour une seule utilisation. Ils sont donc changés à chaque émission/réception.

6.3.1.1 Génération des pseudonymes

Dans la suite, les nœuds du réseau sont indexés par $i = 1, .., n$. L'adresse MAC EUI-64 d'un nœud i est formée selon le procédé décrit par Hossen, Kabir et al. dans l'article [76]. Elle est dénotée a_i. Sa taille vaut m bits. Ici $m = 64$.

Cette valeur est choisie pour correspondre à la taille requise d'une adresse MAC utilisée dans les réseaux IEEE 802.15.4. Néanmoins, Ephemeral peut être employé pour générer des identifiants de tailles arbitraires. La fonction qui transforme a_i en un pseudonyme est appelée F. Elle est définie par la formule suivante :

$$\begin{aligned} F(a_i, \mathrm{IV}_t, K_t) &= a_i \oplus (E_{K_t}(\mathrm{IV}_t) \bmod 2^m) \\ \mathrm{IV}_t &= g(\mathrm{IV}_{t-1}), \end{aligned}$$

E est un algorithme de chiffrement par bloc dont les blocs ont une taille supérieure ou égale à ℓ. Pour l'évaluation du schéma, nous avons choisi un AES en mode compteur. Néanmoins, celui-ci peut être remplacé par n'importe quel ℓ-bit algorithme de chiffrement par bloc léger.

Pour générer et vérifier les pseudonymes, les nœuds ont tout d'abord besoin de K_t mais également d'un ensemble d'IV. Ces données représentent les entrées de notre algorithme de chiffrement.

Pour tout $t > 0$, le vecteur d'initialisation IV_t est composé de deux blocs : $\mathrm{IV}_t = R||\mathrm{cpt}_t$. Le premier bloc R est formé de ℓ_1 bits. Ces bits sont générés aléatoirement, spécifiques au nœud i et mis à jour périodiquement. R peut être distribué à chaque nœud du réseau pendant la phase d'association ou généré par le nœud lui-même.

La seconde partie de l'IV a une taille de $\ell_2 = \ell - \ell_1$ bits. cpt_t est un compteur incrémenté périodiquement ou sur évènement suivant la planification choisie.

La fonction de transition g qui permet d'obtenir la nouvelle valeur de IV_t en fonction de la valeur précédente IV_{t-1} est définie par :

$$g(\mathrm{IV}_{t-1}) = R||(\mathrm{cpt}_{t-1} + 1),$$

où $||$ représente la concaténation et $\mathrm{cpt}_0 = 0$.

Une valeur de ℓ bits est ainsi obtenue en sortie de E. Si ℓ est supérieure à la taille m de l'identifiant que l'on souhaite remplacer (ici l'adresse MAC), m bits de ℓ sont sélectionnés. On pourra, par exemple, ne garder que les m LSB bits.

Enfin, une fonction XOR est effectuée avec l'adresse a_i pour obtenir le pseudonyme.

Afin de générer les pseudonymes nécessaires pour les adresses MAC source et destination, un nœud i doit stocker la valeur de R ainsi que la valeur courante de cpt_t le concernant mais également pour chacun des voisins $j(j \neq i)$. Les valeurs des R pour chaque nœud j sont différentes et partagées. Les valeurs des cpt_t sont calculées indépendamment des autres nœuds évitant un besoin de synchronisation. Les vecteurs d'initialisation sont des valeurs publiques. En revanche, la clé est secrète.

Pour cela, lors de l'association, chaque nœud i récupère de ses q voisins leurs adresses MAC a_j et la valeur R_j correspondante. Une table appelée "table de privacy" contenant q entrées est créée. Elle stocke la valeur courante de cpt_t pour chaque adresse a_j.

Quand le nœud i souhaite communiquer avec son voisin j, il doit tout d'abord générer les pseudonymes pour les adresses MAC source a_i et destination a_j. Pour cela, il va générer son pseudonyme avec F et son matériel nécessaire à la génération des pseudonymes. Il peut ainsi remplacer l'adresse source par ce pseudonyme. Il calcule ensuite le pseudonyme de la destination du prochain saut en utilisant la même fonction F. Il doit alors utiliser le matériel stocké concernant le nœud a_j.

Toutes ces étapes réussies, l'émetteur peut construire la nouvelle trame MAC chiffrée avec LK. Les adresses MAC en clair sont alors remplacées par les pseudonymes. Afin de permettre leur vérification, les valeurs des cpt_t utilisées pour Ephemeral sont incluses dans l'en-tête MAC comme le montre la Figure 6.3.

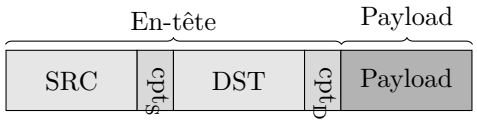


FIGURE 6.3 – Structure d'un paquet Ephemeral.

6.3.1.2 Vérification des pseudonymes

Quand le nœud j reçoit une trame MAC utilisant Ephemeral, il va devoir vérifier si celle-ci lui est destinée ou s'il doit la rejeter. Il va comparer le pseudonyme utilisé comme adresse destination dans la trame reçue et son adresse MAC réelle. La connaissance de la valeur de R et du compteur courant cpt_t associé permet de retrouver l'adresse réelle cachée derrière le pseudonyme.

Pour cela, le nœud j combine la valeur stockée de son R avec la valeur du compteur cpt_D extraite de l'en-tête de la trame Ephemeral afin d'obtenir le vecteur d'initialisation IV_t utilisé par l'émetteur. Il appelle ensuite la même fonction F que celle qu'a utilisé l'émetteur : $F(DST, \mathrm{IV}_t, K_t)$ où DST est le pseudonyme destination inclus dans l'en-tête de la Figure 6.3.

Il vérifie alors si l'adresse obtenue par F correspond à son adresse MAC réelle. Si ce n'est pas le cas, il filtre la trame de manière classique.

Si la trame lui est adressée, il effectue le même travaille sur le pseudonyme source reçu afin de retrouver l'adresse réelle du nœud émetteur. Il déchiffre alors le *payload* MAC à l'aide de la clé LK afin d'accéder au contenu de la trame.

Il identifie ensuite si la trame doit être routée ou si elle lui est destinée grâce à l'adresse IPv6 réelle. Si celle-ci doit être routée, il chiffre alors le *payload* de la trame avec la sécurité MAC. Il va ensuite effectuer le même travail pour fournir les pseudonymes correspondant dans l'en-tête MAC.

Tous les nœuds appartenant au chemin déploient Ephemeral jusqu'à ce que la trame atteigne sa destination finale.

L'utilisation d'une fonction cryptographique et la mise en place d'Ephemeral nécessitent d'observer quelques règles au risque de casser la protection de la vie privée et de permettre à un attaquant de lier les pseudonymes dynamiques à une adresse réelle.

6.3.1.3 Analyse de la protection de la vie privée

Lorsque l'on utilise des pseudonymes, il faut s'assurer que deux nœuds ne génèrent pas le même pseudonyme pour deux adresses différentes. L'occurrence des collisions est donc critique pour assurer l'unicité des pseudonymes et pour empêcher des ambiguïtés pendant le routage. La probabilité que deux nœuds partagent le même pseudonyme est donnée par le paradoxe des anniversaires.

Tant que $n \ll 2^{\frac{m}{2}}$, la probabilité de collision est négligeable. Dans notre cas, $m = 64$. Le nombre de nœuds n dans le réseau est fixé environ à 30.

L'IV_t doit également respecter certaines caractéristiques. En effet, l'utilisation d'une fonction cryptographique comme l'AES rend dangereux la réutilisation d'un même IV_t avec une clé identique. Afin d'assurer la protection de la vie privée, l'IV_t ne doit donc pas être réutilisé sous peine de créer des collisions de pseudonymes voire de permettre à un attaquant de retrouver l'adresse MAC réelle.

Afin de comprendre quand ces collisions peuvent apparaitre et dans quelles conditions les IV_t peuvent être réutilisés, le choix de la fonction g utilisée dans la mise à jour de l'IV_t mais également la structure de ce dernier sont fondamentaux.

Zenner étudie dans [117] les différentes méthodes de génération d'IV, les possibilités dans la mise à jour mais également les problèmes de collisions lorsque ceux-ci doivent être mis à zéro. Le but de son article est de fournir à un designer une aide dans le choix de la construction de l'IV suivant les besoins de l'application. L'auteur détaille quatre types de générateurs de nombre à usage unique.
Dans le premier cas, le nombre est obtenu grâce à un générateur de nombre aléatoire. Néanmoins, cette solution nécessite une taille importante pour le nombre généré. De plus, il y a un risque de collisions probables et donc nécessite un bon générateur pour les éviter, ce qui représente encore un challenge.
Dans le second cas le plus répandu, l'IV est un compteur incrémenté à chaque mise à jour. Cette solution est la plus efficace tant que le compteur ne boucle pas sur une valeur déjà utilisée. Dans ce cas comme lors de l'initialisation, il est nécessaire de "mettre à zéro" la valeur du compteur. Deux voies sont exploitables. Il est possible de mettre à zéro le compteur à une valeur aléatoire. Cette solution nécessite l'intervention d'un générateur de nombre aléatoire et donc subit les problèmes liés à ce dernier. Dans la deuxième possibilité, la mise à jour se fait grâce à des points de *reset* stockés dans la mémoire non volatile du nœud afin d'empêcher les collisions. Néanmoins, cela nécessite de pouvoir y accéder facilement.
La troisième solution appelée *m*ixed solution 1 consiste à concaténer une valeur aléatoire et un compteur. A chaque changement d'IV, le compteur est incrémenté et la partie aléatoire est de nouveau générée. Cette solution faisant de nouveau intervenir un générateur de nombre aléatoire, elle subit les mêmes inconvénients. Néanmoins, l'utilisation d'un compteur réduit le risque de collisions. Afin de pallier au problème du générateur lors de la mise à jour de l'IV, une solution appelée *m*ixed solution 2 a été introduite. Elle correspond à la concaténation de ℓ_1-bit aléatoires avec un compteur de taille ℓ_2-bit comme pour la solution *m*ixed solution 1. Néanmoins, lors de la mise à jour, seul le compteur est incrémenté, la valeur aléatoire reste identique tant que le compteur n'a pas atteint sa valeur maximale. Lors de la mise à zéro, le compteur prend la valeur 0 et la partie aléatoire est de nouveau générée. Lorsqu'il est impossible d'accéder à la mémoire pour mettre en place la solution utilisant les points de *reset*, cette solution est celle qui permet d'obtenir la plus petite probabilité de collisions pour une taille d'IV la plus petite. Notre fonction g pour la mise à jour de notre IV correspond à cette solution.

La formule suivante est donnée pour caractériser l'occurrence d'une collision :

$$\ell \geq \log_2 \left(n \cdot \frac{\alpha^2 - \alpha}{2 \cdot p_{max}} \right)$$

La probabilité d'une collision, dans le cas d'un *reset* de IV_t, est inférieure à p_{max}. α représente le nombre de *reset* de IV_t.

Prenons l'exemple de la planification d'Ephemeral périodique. Imaginons que cpt_t est incrémenté toutes les 60s. Imaginons également que la taille de cpt_t soit $\ell_2 = 8$ bits. Avec ce cas d'usage, R doit être mis à jour toutes les 256 minutes maximum.

Supposons que le WSN déployé ait une durée de vie de 10 ans. 20 532 mises à jour auront alors lieu.

Si on considère $n = 30$ nœuds dans le réseau, la probabilité de collision maximale est donc d'environ $p_{max} \approx 2^{-95}$. Ce qui reste assez faible.

Un autre problème apporté par Ephemeral concerne les cpt_t. Ces compteurs sont envoyés en clair dans l'en-tête MAC afin de permettre la vérification des pseudonymes. Un attaquant peut tenter d'utiliser les valeurs des cpt_t pour identifier un nœud.

En effet, chaque nœud possède une valeur de cpt_t différente. cpt_t peut donc être utilisé comme identifiant unique. De même, quand un nœud rejoint un réseau déjà établi, le compteur de ce nouveau nœud n'est pas synchronisé (ou proche) des valeurs des cpt_t utilisées dans le WSN. Cette désynchronisation des compteurs permet à un attaquant d'identifier une trame associée à un nœud.

Afin d'empêcher ce type d'attaque, chaque nœud va vérifier si les valeurs des compteurs (stockées et reçues) sont toutes proches. Par exemple, considérons deux nœuds i (cpt_t) et i' (cpt'_t) sans perte de généralités. Si $\mathrm{cpt}_t \geq \mathrm{cpt}'_t + \delta$ pour $\delta \geq 0$, alors la valeur de cpt'_t est mise à jour avec la valeur de cpt_t. L'un des cas les plus extrêmes correspond à assurer $\delta = 0$.

6.3.2 Adéquation d'Ephemeral avec les spécifications listées dans 6.2

Ephemeral permet de remplacer les adresses MAC IEEE 802.15.4 source et destination par des pseudonymes dynamiques générés cryptographiquement indépendamment des couches hautes déployées. Il permet de cacher l'activité des nœuds mais également d'empêcher leur suivi ou encore de préserver les identités des nœuds face à de l'écoute passive.

Ephemeral présente l'avantage d'être conçu pour les WSN et les réseaux meshés. Il fonctionne en concordance avec la sécurité MAC.

Un critère de notre cahier des charges concerne la cohabitation avec le mécanisme d'adressage SLAAC. Dans Ephemeral, les adresses IPv6 sont chiffrées par la sécurité MAC et la clé LK. De cette façon, les nœuds peuvent utiliser le protocole SLAAC et leur adresse MAC EUI-64 ainsi que le préfixe contenu dans les trames DIO pour la configuration de leurs adresses IPv6.

Grâce à ce comportement, les adresses IP restent identiques tout au long de la durée de fonctionnement du réseau. Elles peuvent ainsi n'être enregistrées qu'une seule fois dans les tables de routage et réutilisées ensuite par RPL sans nécessiter de mises à jour inutiles de celles-ci. La table des voisins stocke des adresses MAC. Contrairement à l'utilisation d'une fonction de hachage, l'AES en mode compteur permet aux nœuds du réseau de retrouver l'adresse MAC réelle utilisée pour la génération du pseudonyme. Les entrées sont donc réalisées à partir des adresses MAC réelles des nœuds. Les tables des voisins et de routage sont ainsi construites de manière classique avec ces adresses.

Les pseudonymes peuvent être changés sans nécessiter une reconstruction des tables de routage ou du DODAG. Bien que pensé pour être compatible avec RPL et SLAAC, son fonctionnement le rend indépendant du protocole de routage.

Du fait que les identifiants des couches hautes comme les adresses IPv6 ne soient pas impactés par Ephemeral, le nœud peut simplement effectuer le routage comme dans un WSN classique. Comme les tables

de routage ont été enregistrées avec les adresses réelles, il vérifie alors simplement la route dans sa table de routage.

L'utilisation d'un AES présente un autre avantage. En effet, l'AES est déjà implémenté et utilisé pour la sécurité au niveau de la couche MAC. Il est donc possible de réutiliser celui-ci. Il suffit alors de simplement l'instancier. De cette façon, la mise en place d'une fonction cryptographique pour la génération des pseudonymes ne pénalise pas l'image mémoire.

Un autre critère concerne la possible désynchronisation des nœuds. Avec Ephemeral, les horloges des nœuds n'ont pas besoin d'être synchronisées pour autoriser le changement et la vérification de pseudonymes. Les informations contenues dans la trame suffisent pour vérifier les pseudonymes même si un nœud a "loupé" un changement. Les performances ne sont donc pas impactées. Seule une synchronisation des compteurs est prévue pour limiter l'utilisation de ceux-ci comme identifiant unique. Ce fonctionnement permet d'envisager une gestion plus agile du problème de synchronisation. Toutefois, ce critère reste à tester.

Néanmoins, Ephemeral nécessite la génération, le stockage et la diffusion du matériel de l'AES. Il faut que les nœuds connaissent les IV et compteurs utilisés pour la génération des pseudonymes. La création et la vérification des pseudonymes apportent également un surcoût de calcul qu'il est essentiel de quantifier.

De même, la flexibilité d'Ephemeral durant un cycle de vie classique d'un réseau est à tester. Pour tous ces critères, nous avons besoins d'évaluer notre solution par rapport au déploiement d'un réseau de référence.

6.4 Conclusion

Un cahier des charges nous a permis de proposer Ephemeral, une solution de génération de pseudonymes dynamiques pour dissimulation des adresses MAC. Ephemeral permet de dissimuler les adresses IPv6 via l'utilisation de la sécurité à la couche MAC.

Bien qu'en adéquation avec de nombreux critères du cahier des charges, Ephemeral n'est pas une solution parfaite et les pertes de performances apportées par l'utilisation de pseudonymes doivent être évaluées vis-à-vis de celles d'un réseau de référence.

De même, les lacunes identifiées par analyse de MT6D doivent être testées et comparées à celles apportées par Ephemeral. Nous allons par la suite déployer ces trois réseaux et analyser leurs comportements et leurs performances.

Partie 7

Evaluation d'Ephemeral

Dans cette partie, les performances d'Ephemeral vont être étudiées. Une première étude vise à démontrer l'adéquation d'Ephemeral avec le cahier des charges présenté dans la partie précédente. Elle permet de mettre également en lumière les lacunes identifiées théoriquement sur MT6D. La mise en place du simulateur et le scénario type utilisé pour la comparaison de trois réseaux sont présentés. Cette simulation positionne Ephemeral vis-à-vis d'un réseau de référence sans solution de dissimulation des adresses et de MT6D. Nous démontrons que malgré des performances quelque peu dégradées, Ephemeral propose une solution plus adaptée aux réseaux contraints que MT6D. Fort de ce résultat, le déploiement réel d'Ephemeral est donné ainsi que le système d'écoute nécessaire à l'évaluation des performances d'Ephemeral. L'analyse de l'impact d'Ephemeral sur la mémoire, la consommation énergétique et le temps de calcul nous permet de quantifier le coût de déploiement d'une solution de dissimulation des adresses dans un réseau contraint.

7.1 Simulation

Les analyses réalisées sur MT6D et Ephemeral montrent que les critères définis dans le cahier des charges ne sont pas tous respectés. Il est donc essentiel de compléter cette analyse par une phase de test afin de corroborer analyse théorique et déploiement mais également de quantifier ces différences et d'analyser le comportement des solutions lors de cas d'usage concrets.

Le but de cette étude est de fournir une référence de déploiement d'un réseau sans solution de dissimulation d'adresses. Le comportement de ce réseau est alors comparé à un réseau embarquant MT6D et à un autre embarquant Ephemeral.

Nous avons développé une plateforme de simulation nous permettant de définir rapidement différentes topologies et cas d'usage.

7.1.1 Environnement de simulation

Le schéma de principe de la plateforme simulée est donné dans la Figure 7.1. Elle est composée de différents éléments permettant les communications entre un hôte Ethernet et des nœuds du WSN.

L'hôte Ethernet est utilisé pour simuler un client/serveur Internet. Cet hôte envoie et collecte des données provenant du WSN. Une application Java joue le rôle d'interface. Elle ouvre un port sur un *socket* UDP à l'adresse IPv6 fournie par Tunslip.

Tunslip a pour rôle de convertir les paquets IPv6 en données pour le lien série et vice versa afin d'interconnecter le réseau IPv6 Ethernet et le réseau 6LoWPAN. Ce nouveau lien est déployé entre le client IP et le 6BR.

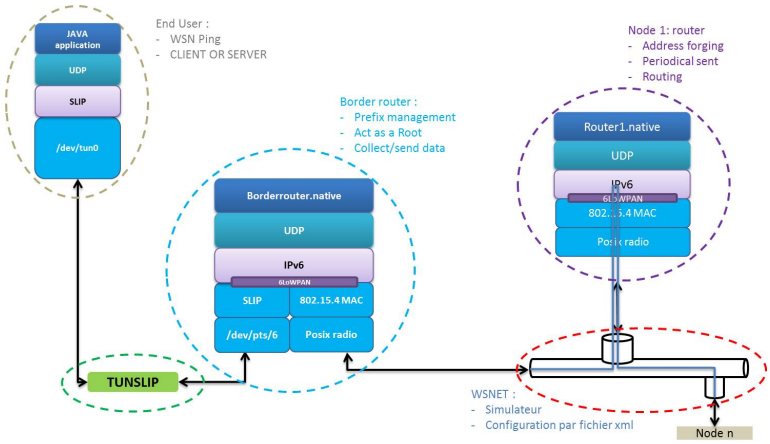


FIGURE 7.1 – Réseau simulé.

Le 6BR assure l'interopérabilité entre les deux piles de communication. Pour cela, il implémente les couches basses de l'IEEE 802.15.4 côté réseau 6LoWPAN et celles du *Serial Line Internet Protocol* (SLIP) côté hôte. Il représente le point d'entrée/sortie du WSN.

Afin de permettre les communications dans le réseau 6LoWPAN, un lien radio est simulé utilisant Posix radio et le simulateur WSNET2.0. La couche radio étant dépendante du matériel et donc de la plateforme, la radio posix remplace la couche radio habituelle (couche Physique). Elle permet la communication avec WSNET en ouvrant des *pipes* avec chacun des nœuds du WSN. Les nœuds du réseau peuvent être soit routeur soit feuille (End Device).

WSNET est un simulateur *event-driven* pour WSN qui permet à un utilisateur d'ajouter des modifications au simulateur sans avoir à modifier le noyau de WSNET. Pour cela, un fichier XML de configuration est utilisé. Dans ce fichier, il est possible de définir les différentes couches de communication, mais également la propagation radio grâce à des modèles proposés. Néanmoins, l'étude que nous souhaitons réaliser se focalise sur le comportement des protocoles de dissimulation d'adresses. Nous souhaitons étudier uniquement l'impact de ces solutions. Nous avons donc configuré WSNET pour une couche Physique parfaite. Seuls les paramètres de propagation ont été utilisés afin de mettre en place un réseau meshé.

Les couches au-dessus de la couche Physique sont fournies par l'OS Contiki. Les nœuds implémentent donc la pile de communication de Contiki 3.0 en mode natif modifiée pour fonctionner avec WSNET. Comparée aux précédentes versions, la distribution 3.0 de Contiki permet l'utilisation de la sécurité MAC essentielle pour le déploiement d'Ephemeral.

7.1.2 Scénario type

Afin que les simulations restent raisonnables en taille et en durée, le réseau de capteurs simulé comporte 5 nœuds organisés selon la topologie de la Figure 7.2. Il est composé d'un nœud 6BR "R".

Il collecte les données provenant des nœuds du WSN et les transmet à l'hôte via Tunslip. Dans l'autre direction, les données envoyées par l'hôte Ethernet sont routées dans le WSN via le 6BR. Il joue donc le rôle de passerelle. Le rôle de nœud *root* est également implémenté dans le 6BR. Situé au sommet du DODAG RPL, il gère, initialise et configure le réseau. Le nœud *root* récupère également le préfixe IPv6 fourni par Tunslip et utilisé pour le processus SLAAC. Il configure alors ses adresses IPv6 puis transfert le préfixe aux nœuds du WSN via l'utilisation des trames DIO de RPL.

Les nœuds "A" et "B" sont des routeurs et "C" et "D" sont des feuilles. Indépendamment de leur rôle, les nœuds sont capables de forger leurs adresses et d'envoyer des données. Les nœuds routeurs peuvent également router les données jusqu'à la destination désirée.

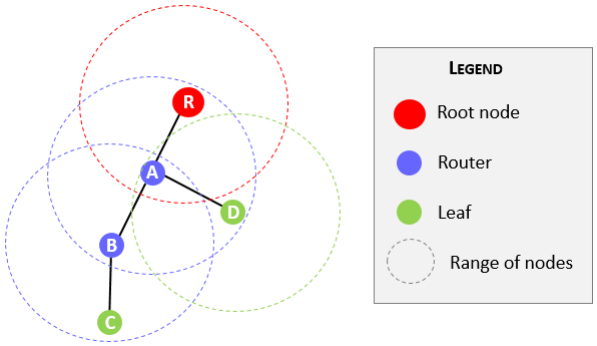


Figure 7.2 – Topologie utilisée.

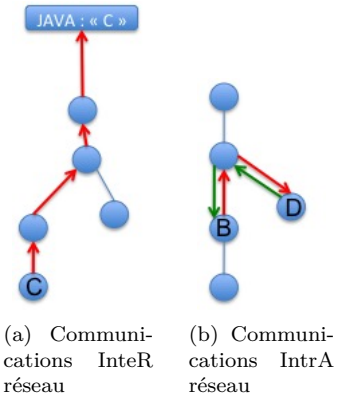
Grâce à notre simulateur et Contiki, trois réseaux ont pu être déployés :

1. Un réseau de référence. Il implémente SLAAC et RPL. La sécurité n'est pas déployée ni aucune solution de dissimulation des adresses.
2. Un réseau avec des pseudonymes IPv6 et MAC selon le schéma MT6D.
3. Ephemeral : Utilisation de pseudonymes pour les adresses MAC et du chiffrement couche MAC.

Le même scénario est reproduit pour chacune des expériences. Le scénario type se déroule en plusieurs phases :

- Les programmes pour l'application Java, Tunslip, WSNET ainsi que chaque nœud du réseau sont lancés simultanément.
- Lors de la première phase, le nœud "R" attend de Tunslip la valeur du préfixe utilisé.
- Dès que "R" a obtenu cette valeur, il configure son adresse IPv6 et se rend visible pour les autres nœuds du réseau.
- Les nœuds du réseau peuvent alors commencer le protocole d'association défini par RPL.
- Dès qu'un nœud a terminé sa phase d'association, il peut commencer à communiquer. Le protocole de maintien des tables de routage de RPL est programmé.
- À la fin de l'expérience, tous les programmes sont terminés simultanément.

Deux types de communications sont étudiés (Figure 7.3).



(a) Communications InteR réseau

(b) Communications IntrA réseau

Figure 7.3 – Communications dans le WSN.

Dans un premier type de communications, les nœuds vont envoyer périodiquement des trames UDP contenant leur identité à l'application Java. Les communications passeront toutes par le nœud *root*. L'application

Java collecte les données reçues et les affiche dans un terminal. Dans l'exemple de la Figure 7.3(a), le nœud "C" envoie à l'application Java une trame contenant son identité. Les nœuds intermédiaires vont router cette trame jusqu'au nœud *root* qui transmet alors la trame à l'application Java. Cette dernière affiche ensuite la valeur de l'identité reçue. Par la suite, ce type de communications sera appelé InteR-network (IR) car elle concerne des communications entre deux réseaux.

Dans un deuxième cas d'usage, les nœuds communiqueront entre eux. Un schéma de client/serveur est établi. Chaque nœud, par exemple le nœud "D" de la Figure 7.3(b), va périodiquement envoyer une demande d'identité (flèche verte) à un nœud du DODAG (le nœud "B"). En réponse, le nœud renverra une trame UDP (flèche rouge) contenant son identité. Les communications seront internes au réseau et il sera possible d'observer le fonctionnement du routage dans ce cas d'usage. On parlera de communications IntrA-network (IA).

Chaque expérience dure 20 minutes. L'émission périodique a lieu toutes les 10s. Pour les expériences avec MT6D et Ephemeral, la périodicité du changement de pseudonyme est fixée à 60s.

Ce scénario nous permet d'observer le réseau de référence lors des phases que sont l'association, le maintien du routage et les communications. Il nous permet d'étudier le comportement des nœuds suivant leur rôle et leur position dans le réseau pendant ces phases. Les données essentielles au routage comme les tables ou les trames de contrôle échangées sont également observables. Enfin, le trafic peut être étudié afin de définir un modèle des communications pour le réseau de référence. Ces différents phénomènes peuvent ensuite être comparés à ceux observés dans les réseaux embarquant les solutions de protection de la vie privée afin de quantifier l'impact de ces solutions mais également les lacunes vis-à-vis du cahier des charges parfait.

L'utilisation d'un simulateur pour cette phase de test et d'analyse offre de nombreux avantages mais également quelques inconvénients.

7.1.3 Analyse comportementale

L'utilisation d'un simulateur pour l'analyse du comportement des solutions de dissimulation des adresses permet un déploiement plus rapide et facile de ces réseaux.

Tout d'abord, la mise en place d'un réseau meshé multi saut est facilité. En effet, lors de déploiement réel, il est compliqué de déterminer comment l'arbre de routage va se former. Les nœuds choisissent leur parent préféré en fonction de métriques radio difficiles à estimer et pas forcément reproductible et réciproque. L'un des moyens les plus simples pour s'assurer d'obtenir un réseau meshé ou une topologie connue est d'éloigner les nœuds entre eux et du nœud *root*. Ainsi en jouant sur la localisation et la distance entre les nœuds il est possible de limiter les nœuds à portée et donc d'obtenir une topologie proche de celle désirée. Néanmoins, suivant l'antenne et donc la portée de celle-ci, il est nécessaire de beaucoup les éloigner. Cet éloignement est très facilement réalisable en simulation. De plus, contrairement à un déploiement réel, il n'y a pas de limite d'espace. Le fichier XML de WSNET permet de déterminer la position d'un nœud mais également sa portée. Ces deux caractéristiques permettent d'établir des liens entre les nœuds et donc mettre en place un réseau meshé.

L'utilisation d'un simulateur permet également une reproductibilité de cette topologie indépendamment de la présence ou non d'une solution de dissimulation des identifiants. Ainsi, il est possible de se focaliser uniquement sur les problèmes liés aux déploiements de ces solutions de protection de vie privée et cela facilite la comparaison avec le réseau de référence. Il n'est également pas nécessaire de mener plusieurs fois la même expérience et de moyenner les résultats obtenus. La reproductibilité rend l'étude plus rapide à mener.

La surveillance du réseau est également plus simple. En effet, alors qu'en réel il est nécessaire de déployer des collecteurs ayant une portée limitée et du matériel (PC, cables USB...) pour récupérer les informations de chaque nœud du réseau, en simulation, tout le trafic peut être collecté en enregistrant les données circulant grâce à WSNET sans limite de portée. Il n'est pas nécessaire de recourir à plusieurs collecteurs ni de devoir par la suite regrouper et trier leurs informations au sein d'un même fichier. De même, la surveillance de chaque nœud ne nécessite pas le déploiement et la connexion de chaque nœud avec un PC. On peut ainsi observer la différence de comportement vis-à-vis du type de nœud mais également de sa position dans l'arbre de routage.

Le dernier avantage du simulateur concerne sa radio configurable. Dans nos expériences celle-ci est parfaite. Il n'y a donc pas de pertes de paquets contrairement à un déploiement réel. Cela permet de comparer le comportement des différentes solutions sans se soucier des aléas de la radio. Si des pertes de paquets se produisent par rapport à ce qui est observé pour la référence, cela est dû à la solution de dissimulation et aux processus supplémentaires engendrés par celle-ci (génération d'un pseudonyme, reconstruction de l'arbre, vérification d'un pseudonyme...).

Néanmoins, l'absence d'une couche Physique a engendré des problèmes avec RPL. En effet, ce dernier utilise les métriques physiques dans le choix du parent lors de la construction de l'arbre de routage. Notre environnement ne permet pas de gérer ces métriques et MT6D nécessite la reconstruction périodique du DODAG. Dans l'implémentation sur Contiki OS de MT6D donnée dans l'Annexe C, nous avons donc dû faire des choix d'implémentation et modifier celle décrite par Preiss, Sherburne et al. dans [115]. Néanmoins, la solution retenue permet de faire fonctionner MT6D avec notre simulateur sans apporter de modifications au comportement attendu ni introduire de surcoût.

7.2 Evaluation

Sur cette même plateforme simulée, nous avons comparé les performances et le comportement d'Ephemeral avec un schéma de référence qui utilise RPL sans pseudonyme et MT6D.

Deux types de métriques ont été choisies pour observer et analyser les réseaux :

- Afin d'analyser en profondeur le comportement de RPL, chaque nœud du DODAG surveille l'évolution dynamique de ses tables. Ces données sont ensuite enregistrées dans un fichier pour traitement. Afin d'obtenir une vision temporelle de l'évolution, un *timestamp* a été ajouté à chaque nouvelle entrée des tables.
- Nous souhaitons évaluer l'impact de chacune des solutions de protection des identifiants sur les performances mais également sur le comportement du réseau. Pour cela, les données brutes échangées sont enregistrées puis analysées grâce à Wireshark. Il sera ainsi possible de connaitre les pertes de paquets dues aux solutions mais également le surcoût de trafic de chacune.

7.2.1 Réseau de référence

Nous avons implémenté le scénario type sur Contiki 3.0. Dans le cas du réseau de référence, aucune sécurité ni solution de protection de la vie privée n'est déployée. Le réseau s'établi grâce à la phase d'association de RPL puis les communications ont lieu. Deux expériences sont menées sur ce réseau : l'une pour les communications IA et l'une pour les IR.

Contiki permet de configurer de nombreuses valeurs relatives aux différents protocoles et notamment pour le routage. Nous avons choisi de ne pas changer ces valeurs et donc d'utiliser les valeurs par défaut.

La métrique par défaut utilisée dans le choix du parent préféré est définie par l'OF. Lors d'un déploiement réel, c'est ETX qui est utilisée. Malheureusement, cette métrique est fournie par la radio. Elle ne peut donc pas être utilisée en simulation. Contiki définie cette valeur comme maximale quand elle n'est pas disponible. C'est pourquoi, dans notre réseau simulé, lors de la phase d'association, si plusieurs nœuds avertissent leur voisin d'un rang similaire, l'émetteur du premier DIO reçu sera choisi comme parent préféré.

Concernant le routage, le mode par défaut de Contiki est le mode *storing*. Il nécessite que chaque nœud du réseau stocke deux tables : la table des voisins et la table de routage. Dans Contiki, les tables ne peuvent excéder par défaut 30 entrées chacune.

Afin de maintenir le routage, RPL définit l'envoi périodique de trames de contrôle. L'intervalle de temps est défini par deux constantes :

- RPL_DIO_INTERVAL_MIN fixé à 4.096s
- RPL_DIO_INTERVAL_DOUBLINGS qui fixe la valeur max entre deux DIO à 17 min 28s environ.

Grâce à l'analyse de la référence lors des expériences, plusieurs comportements ont pu être observés.

Dans nos expériences, le réseau est statique. Dès que la phase d'association est terminée, aucun nœud n'est mobile ni disparait. Ce comportement influe sur les métriques observées.

Tout d'abord, une fois l'association terminée, le DODAG est établi et sa topologie n'est alors pas modifiée. Les deux tables n'évoluent plus. La Figure 7.4 donne les tables observées pour le routeur A situé proche du nœud *root* "R". Le nœud *root* enregistre et stocke tous les chemins de routage présents dans le DODAG. Les routeurs, quant à eux, n'enregistrent que les routes passant par leurs enfants. Enfin, les feuilles ne stockent pas de table de routage. Le nombre d'entrées de chaque table est fixe.

Neighbor Table of Node A				
9/2/2016 18:14:17	IPv6	fe:80::03:02:03:04:05:06:07:0R	MAC	01:02:03:04:05:06:07:0R
	IPv6	fe:80::03:02:03:04:05:06:07:0B	MAC	01:02:03:04:05:06:07:0B
	IPv6	fe:80::03:02:03:04:05:06:07:0D	MAC	01:02:03:04:05:06:07:0D

(a) Table des voisins de A.

Routing Table of Node A			
9/2/2016 18:14:17	aa:aa::03:02:03:04:05:06:07:0C	via	fe:80::03:02:03:04:05:06:07:0B
	aa:aa::03:02:03:04:05:06:07:0D	via	fe:80::03:02:03:04:05:06:07:0D
	aa:aa::03:02:03:04:05:06:07:0B	via	fe:80::03:02:03:04:05:06:07:0B

(b) Table de routage de A.

FIGURE 7.4 – Tables du nœud routeur A.

On peut observer que "A" possède une route pour "D", "B" et "C" qui sont situés plus "bas" dans l'arbre de routage mais n'en possède pas pour "R". En revanche "R" est bien reconnu comme voisin de "A" tout comme "B" et "D". "C" n'étant pas un voisin direct (1-saut) de "A", il n'est pas inscrit dans la table des voisins. Seule une route indique qu'il est nécessaire de passer par "B" pour communiquer avec "C".

Pour le trafic, on observe uniquement 5 trames DIS correspondant aux 5 trames échangées durant la phase d'association pour les 5 nœuds du réseau (1 par nœud). Cette observation valide le comportement attendu de RPL. En effet, dès qu'un nœud a rejoint le réseau, il n'est plus obligé de relancer un protocole d'association. Il connait le préfixe et a pu établir son adresse IPv6, il est donc en mesure de communiquer.

Pour les communications de l'application, en 20 minutes avec une émission périodique de 10s, 120 trames doivent être émises. On observe, pour la communication IR au total 951 trames UDP échangées dans tout le réseau. Comme certaines trames doivent être routées, il est normal d'observer plus de trames que les 120 émises. En prenant en compte la position des nœuds dans le réseau et donc le nombre de routage nécessaire pour atteindre le nœud *root* "R", on calcule un peu plus de 118 trames ce qui, avec les aléas de fin d'expérience, valide les 120 attendues. L'application a donc fonctionnée correctement. Un travail analogue sur les communications IA montre le bon fonctionnement de l'expérience.

Le nombre de trames ICMPv6 échangées durant les 20 minutes d'expériences reste identique indépendamment du type de communications (IR : 376 / IA : 375). En revanche, on observe plus de trames UDP dédiées à l'application dans le cas de communications IA (1849) que dans le cas IR (951). Cette différence est due au fonctionnement de l'application. Dans le cas des communications IA, le nœud doit d'abord interroger un autre nœud du réseau en lui envoyant une demande d'identité. Cela déclenche une réponse UDP contenant cette identité. On observe donc environ deux fois plus de trames UDP pour l'application IA que pour IR.

Lors des communications, la tendance observée dans le réseau de référence est que les trames périodiques de l'application sont plus nombreuses que celles du maintien des tables de routage. Ce comportement est désiré afin de limiter la consommation énergétique nécessaire pour les phases de gestion du réseau. En effet, lorsque le réseau est statique, RPL limite les échanges nécessaires au maintien des tables.

7.2.2 MT6D

Afin d'analyser MT6D, notre contribution a été double.

Tout d'abord, le schéma MT6D a été adapté de Contiki 2.7 à Contiki 3.0. Aucun mécanisme de sécurité n'est déployé. Ensuite, contrairement à leur cas d'usage, nous avons déployé MT6D en mode compteur sur un réseau multi sauts afin d'analyser les performances dans le routage. Les adresses sources et destinations MAC et IPv6 ont été modifiées afin d'utiliser des pseudonymes. Le 6BR implémente également la solution de dissimulation des adresses. Cette deuxième contribution a également permis de comprendre les mécanismes et

protocoles en place ainsi que les implications d'une utilisation de pseudonymes. De nombreuses modifications ont dues être faites afin de permettre le bon fonctionnement du réseau. Le détail de l'implémentation sur Contiki et les problèmes rencontrés sont donnés dans l'Annexe C.

Enfin, afin de permettre une analyse plus rapide et efficace, les adresses ont été configurées de manière à faire apparaitre l'adresse réelle du nœud même lorsque celui-ci a pris un pseudonyme. Il est ainsi plus facile de suivre et regrouper les informations concernant un nœud pour l'analyse. Par exemple, si le nœud "B" possède l'adresse : fB :0B :10 :CA :60 :04 :BC :E4 après changement de pseudonyme, il aura l'adresse : fB :0B :C1 :94 :E3 :1A :B2 :94. Le deuxième octet à 0B est conservé pour indiquer que l'adresse est celle du nœud "B".

Avec MT6D, le changement de pseudonymes est vu comme l'ajout d'un nouveau nœud. Les nœuds effectuent donc de nouvelles associations ce qui entraine l'ajout dans les tables de routage d'une nouvelle entrée. Or, notre réseau est statique. Ce qui, pour la référence, n'implique aucun ajout ou suppression de nœud dans le fonctionnement attendu. Le comportement observé diffère de celui de la référence où lorsque l'association est terminée, le DODAG tout comme les tables n'évoluent plus.

Neighbor Table of Node A				
18/2/2016 16:01:22	IPv6	fe:80::fb:0R:04:ff:fe:07:08:00	MAC	f9:0R:04:ff:fe:07:08:00
	IPv6	fe:80::fb:0D:4b:ee:23:2e:ce:36	MAC	fb:0D:4b:ee:23:2e:ce:36
	IPv6	fe:80::fb:0B:10:ca:60:04:bc:e4	MAC	fb:0B:10:ca:60:04:bc:e4
	IPv6	fe:80::fb:0D:4d:04:b1:3f:1c:47	MAC	fb:0D:4d:04:b1:3f:1c:47
	IPv6	fe:80::fb:0B:c1:94:e3:1a:b2:94	MAC	fb:0B:c1:94:e3:1a:b2:94
	IPv6	fe:80::fb:0D:56:77:8c:55:76:50	MAC	fb:0D:56:77:8c:55:76:50
	IPv6	fe:80::fb:0B:4b:ee:23:2e:cc:36	MAC	fb:0B:4b:ee:23:2e:cc:36
	IPv6	fe:80::fb:0D:65:bb:80:2c:e6:41	MAC	fb:0D:65:bb:80:2c:e6:41
	IPv6	fe:80::fb:0B:4d:04:b1:3f:1a:47	MAC	fb:0B:4d:04:b1:3f:1a:47

(a) Table des voisins de A pour MT6D.

Routing Table of Node A			
18/2/2016 16:01:22	aa:aa::fb:0B:4d:04:b1:3f:1a:47	via	fe:80::fb:0B:4d:04:b1:3f:1a:47
	aa:aa::fb:0C:f0:4d:8d:2a:28:77	via	fe:80::fb:0B:4d:04:b1:3f:1a:47
	aa:aa::fb:0D:65:bb:80:2c:e6:41	via	fe:80::fb:0D:65:bb:80:2c:e6:41
	aa:aa::fb:0B:4b:ee:23:2e:cc:36	via	fe:80::fb:0B:4b:ee:23:2e:cc:36
	aa:aa::fb:0C:65:bb:80:2c:e7:41	via	fe:80::fb:0B:4b:ee:23:2e:cc:36
	aa:aa::fb:0D:56:77:8c:55:76:50	via	fe:80::fb:0D:56:77:8c:55:76:50
	aa:aa::fb:0C:56:77:8c:55:78:50	via	fe:80::fb:0B:c1:94:e3:1a:b2:94
	aa:aa::fb:0D:4d:04:b1:3f:1c:47	via	fe:80::fb:0D:4d:04:b1:3f:1c:47
	aa:aa::fb:0B:c1:94:e3:1a:b2:94	via	fe:80::fb:0B:c1:94:e3:1a:b2:94
	aa:aa::fb:0B:10:ca:60:04:bc:e4	via	fe:80::fb:0B:10:ca:60:04:bc:e4
	aa:aa::fb:0D:4b:ee:23:2e:ce:36	via	fe:80::fb:0D:4b:ee:23:2e:ce:36
	aa:aa::fb:0C:4d:04:b1:3f:1b:47	via	fe:80::fb:0B:10:ca:60:04:bc:e4
	aa:aa::fb:0D:c1:94:e3:1a:b4:94	via	fe:80::fb:0D:c1:94:e3:1a:b4:94
	aa:aa::fb:0B:52:14:9b:e4:6b:95	via	fe:80::fb:0B:52:14:9b:e4:6b:95
	aa:aa::fb:0C:4b:ee:23:2e:cd:36	via	fe:80::fb:0B:52:14:9b:e4:6b:95
	aa:aa::fb:0C:c1:94:e3:1a:b3:94	via	fe:80::fb:0B:cc:2b:36:8a:cf:a3
	aa:aa::fb:0D:10:ca:60:04:be:e4	via	fe:80::fb:0D:10:ca:60:04:be:e4
	aa:aa::fb:0B:cc:2b:36:8a:cf:a3	via	fe:80::fb:0B:cc:2b:36:8a:cf:a3

(b) Table de routage de A pour MT6D.

FIGURE 7.5 – Tables du nœud routeur A pour MT6D.

Comme pensé lors de l'analyse théorique, à chaque changement de pseudonymes, les tables utiles au routage vont donc grossir. Elles seront totalement reconstruites à chaque changement de pseudonymes même si le réseau n'évolue pas. La Figure 7.5 montre un exemple de tables pour le nœud routeur "A". Contrairement à ce qui a été observé sur le réseau de référence, la table de routage ne possède pas que 3 entrées mais 18 entrées. De même pour la table des voisins. Néanmoins, toutes ces entrées ne concernent que les 3 nœuds "B", "C", "D" comme c'était le cas pour la référence. La différence tient au fait que 15 des 18 entrées ne sont plus valides car les pseudonymes sont périmés.

Lorsque le maximum d'entrées est atteint, les voisins les plus anciens sont supprimés afin de libérer de la place pour les nouveaux. Le même procédé est utilisé sur la table de routage. Un élément de la table de routage n'est donc supprimé que si la taille maximale est atteinte. Le fonctionnement classique de RPL n'est donc plus respecté et l'un des critères du cahier des charges idéal n'est pas suivi.

Ce comportement impacte également les trames de contrôle émises dans le réseau. En effet, afin de reconstruire le réseau, le protocole RPL est lancé à chaque nouveau pseudonyme. Contrairement à la référence, le nombre de trames DIS passe de 5 à 85. Comme le changement de pseudonymes apparait toutes les 60s et que l'expérience dure 20 minutes, les 85 trames DIS correspondent aux 20 changements de pseudonymes pour les 4 nœuds du réseau ajoutés au 5 DIS de la première association. On observe donc bien que MT6D comme attendu relance le protocole d'association entrainant une augmentation des trames de contrôle.

Le nombre de trames ICMPv6 utilisées pour le contrôle et le maintien du routage est donc augmenté. La diffusion et le partage des pseudonymes apportent un surcoût. Ce critère du cahier des charges n'est donc pas respecté.

Enfin, ce comportement du réseau impacte les communications utiles à l'application. En effet, lorsqu'un nœud procède à l'association, il ne peut communiquer tant que celle-ci n'est pas terminée. Ainsi, au lieu des 120 trames d'application attendues en 20 minutes d'expérience, on n'observe pour les communications IR uniquement 102 trames échangées. Le scénario est donc modifié par rapport à la référence et la qualité de service est réduite. Cet impact est encore plus grand pour les communications IA où au lieu des 120 trames on observe une valeur proche des 90. Cette perte plus importante est due au schéma client/serveur. En effet, si la trame UDP perdue à cause de la reconstruction du DODAG est celle de demande d'identité, la requête n'étant jamais reçue par le nœud, aucune réponse ne sera émise. Il y a donc une perte plus importante.

7.2.3 Ephemeral

Nous avons implémenté Ephemeral sous Contiki 3.0. La sécurité MAC a été mise en place. Le détail de l'implémentation est donné en Annexe D. Un exemple de fonctionnement est également donné dans cette annexe.

La gestion des pseudonymes est implémentée à la couche lien dans le noyau de Contiki et est transparente à un développeur d'applications.

Du fait qu'Ephemeral permette l'enregistrement des tables de routage avec les adresses MAC et IPv6 réelles des nœuds et ce même lors d'un changement de pseudonyme, les tables sont donc identiques à celles obtenues pour le réseau de référence. Ainsi, dès l'association terminée, les tables n'évoluent plus. Les tables ne sont pas retouchées. Ephemeral respecte le critère imposé par notre cahier des charges.

De même, il n'est pas obligatoire de procéder à de nouvelles associations. On observe pour Ephemeral uniquement les 5 trames DIS de l'association de départ. Ephemeral conserve bien le comportement de RPL.

En ce qui concerne les trames utiles à l'application, Ephemeral impact la qualité de service. Pour les communications IR, Ephemeral ne nuit pas aux performances de l'application. On observe bien les 120 trames attendues en 20 minutes. En revanche, pour IA, seuls 112 échanges ont eu lieu au lieu des 115 observés pour la référence. Cette différence est due à deux facteurs. Tout d'abord, Ephemeral introduit des calculs supplémentaires lors de la génération et de la vérification des pseudonymes qui augmentent le temps de traitement d'une trame nécessaire. De plus, Contiki ne possède pas de mécanisme de gestion des queues pour les paquets entrants.

Dans le cas des communications IA, contrairement aux IR, les routeurs doivent gérer plus de paquets entrants. Ils doivent donc gérer plus de trames entrantes/sortantes mais également leurs propres trames d'applications. Le temps de traitement va donc influencer la réussite du routage. Pendant que le nœud gère les calculs supplémentaires, les paquets entrant sont ignorés ce qui induit des pertes.

Ce problème s'observe également sur les phases de maintien du routage. Comme de nombreuses trames de contrôle doivent être traitées et que le nœud peut encore être en phase de traitement d'une trame d'application au moment où il reçoit des trames de contrôle, il existe des pertes. Comme pour les trames UDP, la différence entre la référence et Ephemeral s'observe surtout dans le cas des communications IA. Ainsi, seuls 114 DIO sont échangés au lieu des 133. De même, pour les DAO on passe de 237 à 166. La différence est plus importante car les DAO doivent être routés. Si l'un d'entre eux n'est pas traité, il ne sera pas non plus routé d'où une plus grande différence. Ces pertes peuvent mener à des topologies non optimales ou à des besoins de relancer le protocole de maintien du routage plus fréquemment dans des réseaux où la topologie évolue souvent. Une mauvaise topologie peut entrainer des pertes sur l'application. Il serait intéressant de tester notre solution avec un autre OS qui possède une gestion plus agile des trames entrante/sortante afin de tester l'impact réel du temps de calcul supplémentaire sur la perte de paquets. Il serait également intéressant de mesurer ce temps de calcul supplémentaire. Pour cela, il est obligatoire de procéder à un déploiement réel.

Contrairement à la référence, on n'observe donc pas avec Ephemeral une indépendance vis-à-vis du type de communications pour le nombre de trames ICMPv6. Néanmoins, le ratio entre trames UDP et trames ICMPv6 est approximativement le même que pour la référence. On conserve donc bien le comportement de RPL et son mécanisme de réduction de la consommation d'énergie.

L'un des critères du cahier des charges concerne la diffusion et le partage des pseudonymes mais également du matériel nécessaire. Nous avons vu que MT6D apporte un surcoût non négligeable pour le partage de ces pseudonymes. De par son fonctionnement, Ephemeral permet de vérifier et de générer les pseudonymes sans avoir à les propager. Néanmoins, il nécessite la connaissance du matériel utilisé par l'AES et propre à chaque nœud. Le détail de l'implémentation des différentes caractéristiques est donné dans l'Annexe D.

Tout d'abord, le nœud a besoin de connaitre l'IV utilisé pour la génération du pseudonyme mais également de propager le sien. Une adaptation du protocole RPL a été effectuée afin de diffuser les IV. Le nouveau format des trames DIO/DAO permet à un nœud d'intégrer la valeur de son R dans ces trames. Ainsi, grâce à RPL, Ephemeral n'introduit pas de trafic supplémentaire pour la propagation du matériel. Néanmoins, la taille des trames DIO/DAO est impactée. Dans le cas implémenté sur Contiki, un surcoût de 17 octets est à considérer.

De plus, pour la vérification des pseudonymes, les compteurs doivent être ajoutés à l'en-tête MAC. Cela apporte encore un surcoût sur la taille de la trame. Ces trames de taille plus importante demandent plus d'énergie pour être émises. Un déploiement en simulation ne permet pas d'avoir de réel retour sur ces performances. Il est donc essentiel de déployer le réseau en réel.

Enfin, le matériel de génération des pseudonymes des voisins et du nœud lui-même doit être stocké. Cela peut entrainer une consommation mémoire importante qu'il est essentiel de quantifier.

7.2.4 Synthèse

Cette section permet de comparer les pertes dues aux deux solutions de protection des identifiants vis-à-vis de la référence.

Experience		UDP	ICMPv6	DIS	DIO	DAO
Référence	IR	951	376	5	135	236
	IA	1849	375	5	133	237
MT6D	IR	816	1163	85	355	723
	IA	1446	1120	81	351	688
Ephemeral	IR	952	358	5	133	220
	IA	1800	285	5	114	166

Table 7.1 – Nombre de trames en 20 min.

Pour les trois réseaux, nous avons observé le trafic. Deux types de trames sont échangées : UDP pour l'application et ICMPv6 pour le management du réseau. La Table 7.1 résume le trafic observé lors des communications IA et IR.

Dans le pire des cas, MT6D augmente le nombre total de trames échangées de 49%. Ephemeral, quant à lui, apporte une différence de 6.25% principalement dues aux pertes expliquées précédemment.

En ce qui concerne la qualité de service, dans le cas de MT6D, la perte de paquets UDP par rapport au réseau de référence atteint 21,8% contre 2.6% pour Ephemeral.

La Figure 7.6 montre le ratio en pourcentage entre les trames de contrôle RPL et les trames UDP pour les communications IR (un raisonnement analogue peut être mené sur les communications IA). Dans le cas de la référence, cette figure permet d'illustrer ce qui est attendu comme comportement dans le réseau. Les phases de contrôle et de management du réseau ne doivent pas occuper trop de temps vis-à-vis des phases dédiées à l'application. On peut voir que ce ratio doit être environ de 70/30. Pour Ephemeral, malgré les pertes de paquets, le comportement attendu est présent avec un ratio proche de celui observé pour la référence.

Pour MT6D, le comportement du réseau diffère fortement. Le nombre de trames de contrôle approche celui des trames d'application. C'est environ 3 fois plus que pour la référence. Ce comportement est dû à la nécessité de reconstruire complètement le DODAG après chaque génération de nouveau pseudonyme. Les performances du réseau sont alors perturbées.

Pour conclure, grâce à la mise en place du simulateur, les trois réseaux ont pu être comparés.

Figure 7.6 – ICMPv6 vs UDP.

MT6D modifie le fonctionnement de RPL en modifiant les tables de routage. Il nécessite un surcoût de trames de contrôle pour la propagation des pseudonymes nuisant à la qualité de service.

Ephemeral apparait alors comme une solution de dissimulation des adresses adaptée aux WSN, qui est compatible avec RPL et SLAAC. Les calculs supplémentaires nécessaires à la vérification et la génération de pseudonymes induisent des pertes de paquets nuisant à la qualité de service. Toutefois, il améliore de 16% la qualité de service par rapport à MT6D. De plus, il ne nécessite pas de surcoût de trafic pour la propagation des pseudonymes. En revanche, le format et donc la taille des trames sont modifiés pour permettre la vérification et la génération des pseudonymes. Cette différence impacte l'énergie consommée pour l'émission d'une trame qu'il faut quantifier. De même, la mémorisation du matériel nécessite une occupation mémoire qu'il faut mesurer. Enfin, l'impact des calculs supplémentaires doit être identifié afin de décorreler les pertes dues au problème de gestion des queues de Contiki et l'utilisation d'Ephemeral.

Dans la suite, nous avons voulu évaluer Ephemeral in situ à l'échelle d'un étage (*smart office*).

7.3 Déploiement

Un déploiement réel a été réalisé avec Ephemeral afin d'observer l'empreinte sur des nœuds contraints. Nous avons également évalué le comportement de cette nouvelle technique en situation réelle.

7.3.1 Environnement

Nous avons déployé un réseau réel dans lequel les nœuds de capteurs Openmote embarquent Ephemeral comme solution de dissimulation des adresses source et destination. Nous avons également mené des expériences similaires sur un réseau de référence sans solution de protection des identifiants mais avec la sécurité MAC établie. Nous avons ainsi pu comparer les comportements des deux réseaux et estimer le coût de déploiement d'Ephemeral.

Le réseau est composé d'un nœud *root*, de nœuds routeurs et de nœuds feuilles. 20 nœuds et 1 6BR ont été déployés au sein d'un étage de bâtiment comme le montre la Figure 7.7. Les ronds rouges représentent les capteurs. Le nœud possédant l'adresse F480 et positionné dans la pièce 333 joue le rôle de 6BR et de *root*.

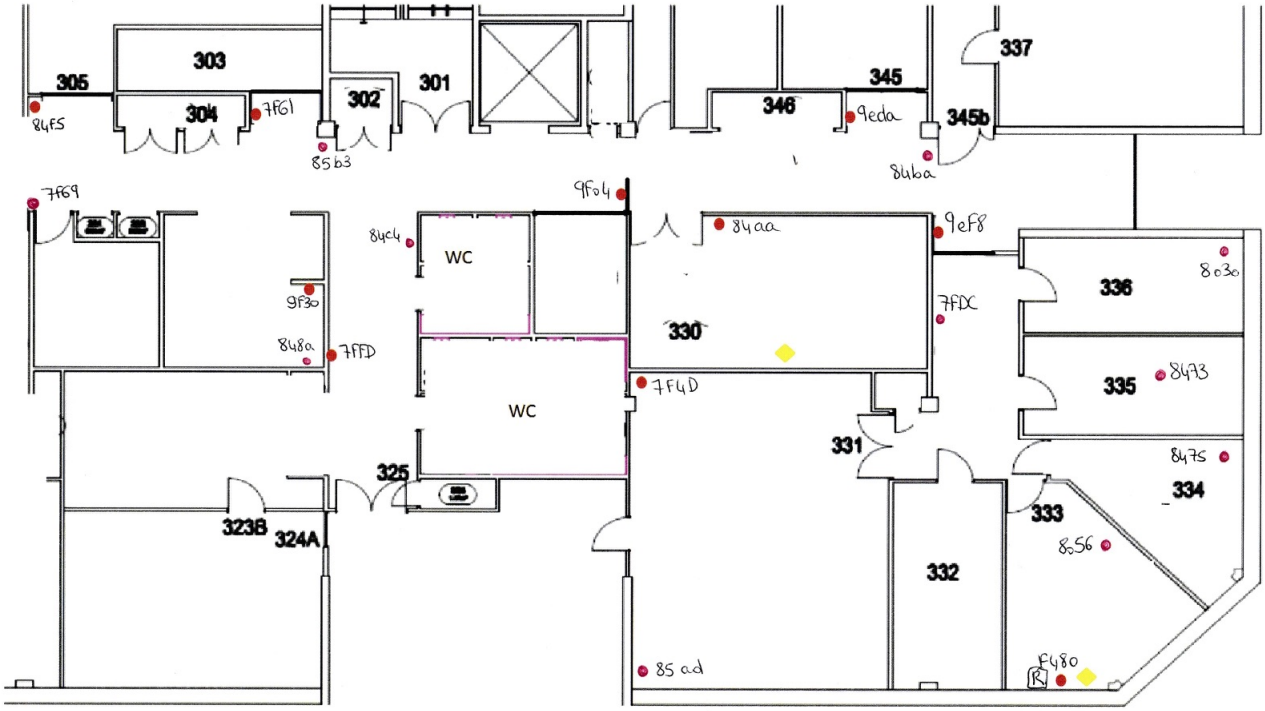


Figure 7.7 – Smart office.

La topologie peut évoluer avec l'ajout ou la suppression de nœuds mais également la prise en compte de la mobilité. Les communications étudiées sont des communications IR dans un réseau de capteurs multi sauts.

Différentes expériences sont menées avec ce réseau afin d'étudier les comportements des nœuds lors de phases de vie différentes. Dans la suite, nous allons regarder également l'impact d'Ephemeral sur de nombreux critères de performances.

7.3.2 Outils de surveillance du réseau

Afin d'obtenir une analyse complète du comportement du réseau lorsque celui-ci implémente Ephemeral, les outils de surveillance déployés précédemment dans la partie 4 sont mis en œuvre. Dans le Figure 7.7, les losanges jaunes correspondent aux différents collecteurs déployés. Les collecteurs interceptent les trames échangées dans le WSN dans leur zone de portée. Wireshark est utilisé pour l'analyse des trames collectées.

Lors de la première expérience, seul un collecteur proche du 6BR a été déployé. Pour cela, le collecteur basé sur le matériel Wismote décrit dans la partie 4 a été utilisé.

Les trames arrivant sur le nœud *root* sont collectées. Certains nœuds du réseau sont également à portée. L'analyse des trames de contrôle arrivant au nœud *root* permet de s'assurer que l'association s'est bien réalisée. Les communications étant routées jusqu'au nœud *root*, lorsque celles-ci se déroulent correctement, le collecteur permet de les analyser.

En revanche, à cause de la faible portée de ce collecteur, tout le trafic dans le réseau ne pouvait être observé. Comme nous voulions analyser le comportement des nœuds par rapport à leur position dans l'arbre de routage mais également connaitre combien et où apparaissaient les pertes de paquets, les informations collectées n'étaient pas suffisantes. Ce collecteur ne pouvait nous permettre d'observer le comportement du réseau.

Le déploiement d'un seul collecteur n'est donc pas suffisant pour évaluer correctement le comportement du réseau lorsqu'Ephemeral est déployé. Il est nécessaire d'avoir une vision plus complète du déploiement.

Pour cela, un deuxième collecteur a été déployé grâce à un Raspberry Pi et d'une antenne plus puissante. Celui-ci a été placé au milieu du réseau. Ainsi tout le réseau est couvert. Néanmoins, l'utilisation de deux collecteurs nécessite de rassembler les trames collectées en un seul fichier, trier les trames en doublons et effectuer quelques mises en forme pour une analyse.

Maintenant, les trames UDP et ICMPv6 utilisées dans le réseau sont accessibles par écoute passive. On peut analyser les pertes de paquets, les trames non routées ou encore les chemins utilisés. Néanmoins, il manque encore des informations pour obtenir une vision complète du comportement. En effet, lors d'une perte de paquets, deux causes sont possibles : soit le média a subi des interférences et la trame émise n'a jamais été reçue par le nœud ciblé, soit c'est le nœud lui-même qui n'a pas traité la trame pour ensuite la router. Dans ce cas, le manque de traitement peut venir d'Ephemeral. Il est donc nécessaire de connaitre la cause de la perte de paquet. Il faut donc déployer une surveillance locale.

L'un des moyens est d'enregistrer pour chaque nœud les trames brutes reçues. Ainsi, si une trame est reçue mais non routée ou traitée, on peut déduire que le nœud n'a pas fini le processus. Les tables de routage sont également enregistrées de manière identique à ce qui a été déployé en simulation. Pour cela, les nœuds doivent être reliés chacun à un PC ce qui est compliqué pour notre déploiement.

Ne possédant pas le matériel nécessaire, nous avons donc choisi de mener différentes expériences et de déplacer le système de surveillance sur différents nœuds suivant leur rôle mais également leur position dans le réseau. Certains nœuds du réseau sont donc surveillés afin d'analyser les changements qu'implique l'emploi de pseudonymes sur les tables de routage et de voisinage mais également pour collecter les trames brutes reçues. Une surveillance locale est donc déployée. L'impact sur le nœud lui-même peut être observé. Ce choix pose problème. En effet, les mêmes conditions pour un déploiement réel ne sont pas aisément reproductibles et les pertes de paquets n'apparaissent pas de manières identiques.

En revanche, cette surveillance permet d'obtenir une première idée du comportement du réseau et de valider la cohabitation d'Ephemeral avec les protocoles existants.

7.3.3 Estimation des performances d'Ephemeral

La sécurité et la confidentialité ont un coût qu'il convient d'évaluer. Dans cette partie, nous avons donc évalué l'impact d'Ephemeral sur la vitesse de génération et de vérification d'un pseudonyme mais également l'empreinte mémoire de notre solution et enfin la consommation d'énergie associée. Ces performances ont été mises en relation avec celles obtenues pour un réseau de référence sans solution de dissimulation des adresses mais où la sécurité MAC est activée.

Tout d'abord, Ephemeral ajoute des étapes de calcul pour l'émission et la réception d'une trame qui peuvent être limitantes. Nous avons vu en simulation que cela entrainait des pertes de paquets. Par rapport au réseau de référence, les étapes supplémentaires concernent principalement le calcul d'AES pour générer les pseudonymes ou les vérifier. Une autre partie concerne la taille plus importante des trames à générer avec l'ajout des en-têtes et des options supplémentaires ainsi que son envoi.

La plateforme Openmote ne possède pas d'AES *hardware*. C'est donc une version logicielle plus lente qui est utilisée pour la sécurité. Néanmoins, l'AES est appelé et instancié dans la couche MAC, couche très basse du modèle OSI. Le temps d'exécution d'un AES en mode compteur seul est donc premièrement observé. Ce temps correspond à la génération d'un pseudonyme ou à sa vérification. Il comprend la lecture des champs dans la table de pseudonymes afin d'extraire le matériel associé nécessaire suivi du calcul de l'AES et enfin du XOR avec l'adresse de départ.

Lors de l'émission, il est nécessaire d'effectuer 2 fois l'AES : un pour le pseudonyme source et un pour le pseudonyme destination. En réception, l'AES peut être effectué plus de 2 fois suivant le nombre d'entrées de la table de pseudonyme et la position, dans celle-ci, du matériel de pseudonyme. En effet, l'implémentation choisie d'Ephemeral oblige le nœud récepteur à vérifier le pseudonyme source reçu avec chaque matériel stocké jusqu'à obtenir la bonne adresse ou ne pas obtenir de réponse positive. Si le pseudonyme reçu concerne la dernière entrée de la table, le nœud devra vérifier toutes les entrées précédentes avant d'obtenir la bonne correspondance.

Ainsi, dans notre implémentation actuelle, le nœud récepteur doit vérifier les pseudonymes avec les IV stockés. Pour le pseudonyme destination, le nœud récepteur n'a qu'à vérifier si le pseudonyme fourni dans la trame est lié à son adresse. Il n'a donc à effectuer qu'un seul AES pour la vérification du pseudonyme destination. En revanche, pour la vérification du pseudonyme source, cela peut nécessiter plusieurs calculs d'AES. Dans le pire des cas, la vérification du pseudonyme source peut utiliser jusqu'à 30 AES (nombre d'entrées maximal dans la table du matériel de pseudonyme). Le nœud récepteur devra dans le pire des cas effectuer 31 AES pour la vérification des deux pseudonymes.

Nous allons donc quantifier le coût d'exécution de l'AES.

Pour cela, dans la littérature, deux méthodes sont proposées :

- Avec les *timers* fournis par Contiki.
- Avec un oscilloscope.

D'après Casado et al. dans [118], la méthode utilisant les *timers* de Contiki permet d'obtenir une meilleure précision que la méthode par oscilloscope. Néanmoins, nous avons voulu tester les deux.

Pour cela, le *timer* utilisé est RTIMER. Une lecture de la valeur du *timer* au début puis à la fin de l'AES nous permet d'obtenir un temps d'exécution. Pour l'évaluation avec l'oscilloscope, les GPIO de l'Openmote sont utilisées. La GPIO est mise à 1 au commencement puis mise à 0 à la fin de l'AES.

Une boucle de 100 générations de pseudonymes est réalisée. Les valeurs obtenues par les deux méthodes sont proches. Le temps de génération d'un AES est d'environ 212 µs (211 µs pour la méthode oscilloscope et 213 µs pour la méthode timer). Soit dans le pire des cas 212 * 31 = 6572 µs = 6,6 ms pour la vérification des pseudonymes en réception.

Maintenant, nous avons voulu quantifier le temps complet d'émission et de réception et le surcoût qu'implique Ephemeral. Ainsi, nous pouvons connaitre le temps de traitement complet d'une trame et quantifier l'impact sur la perte de paquets. En effet, nous avons vu dans la partie simulation que lorsqu'un nœud devait traiter trop de trames entrantes (typiquement un nœud routeur proche du *root*), certaines étaient abandonnées car Contiki ne possède pas de mécanisme de gestion des queues. Cela entraine une perte de paquets. Cette évaluation nous permet de connaitre le débit maximal que peut atteindre l'application pour l'émission de ces données. Nous avons également évalué l'envoi et la réception de trames ICMPv6. Le débit maximal des trames ICMPv6 permet de voir si Ephemeral peut être déployé dans un réseau dense où le nombre de trames de contrôle peut être important pour obtenir une topologie optimale.

Le réseau déployé est composé d'un nœud *root* utilisé afin d'établir le DODAG et un nœud en test (TOE). Dans le cas de l'analyse de l'émission seule, le TOE envoie des trames au BR qui les affiche. Lors du test de la réception, un troisième nœud est utilisé. Il a pour but d'adresser au TOE des trames. Le nœud TOE n'envoie alors plus de trames et traite celles reçues du troisième nœud. Les trames passent donc toutes les couches du modèle OSI de la couche Physique à la couche Application.

Des trames RPL et UDP sont émises soit par le TOE soit par le troisième nœud. Des expériences similaires ont été effectuées pour un réseau de référence et avec Ephemeral.

Les coûts d'une émission et d'une réception d'une trame UDP ainsi que d'une trame RPL ont été analysés par les deux méthodes précédentes. Du fait de l'ajout des en-têtes lors de l'utilisation d'Ephemeral, pour un même type de trame, la taille de celle émise par le réseau de référence sera différente de la taille de celle émise par Ephemeral. Nous avons donc également comparé le temps d'émission avec Ephemeral et pour le réseau de référence pour une trame UDP de taille identique. La taille indiquée correspond à une trame complète.

La Table 7.2 donne le résultat des expériences menées.

Type		Ephemeral			Référence		
		Taille	Nb échantillon	Temps (ms)	Taille	Nb échantillon	Temps (ms)
Réception	UDP	86 + 6	404	7.70	86	491	6.30
	DIO	120 + 5	337	10.06	102	434	7.44
Emission	UDP	86 + 6	122	11.37	86	117	9.70
	DAO	81 + 6	1226	11.21	81	1089	9.61
	UDP	92	141	11.37	92	120	9.96

Table 7.2 – Temps de calcul.

Pour les trames UDP de même taille (92 octets), le traitement et l'émission prend 14% de temps en plus avec Ephemeral que pour la référence. Comme la trame fait exactement la même taille, ce surcoût est donc dû uniquement aux calculs d'Ephemeral (génération des pseudonymes) et à la mise sur l'air de 6 octets supplémentaires. Dans le cas d'une réception de trames DIO, l'utilisation d'Ephemeral augmente d'environ 35% le temps de traitement pour une différence de 23 octets (en-tête MAC + champs IV) entre la taille des deux trames ICMPv6.

Néanmoins, lors de notre test, seul un 6BR et un TOE communiquaient. Le TOE n'avait donc pas à router de trames et ne possédait qu'une seule entrée correspondant au 6BR dans sa table de pseudonymes. Le temps mesuré correspond à la vérification de deux pseudonymes (calcul de 2 AES) ainsi qu'au *parse* des en-têtes.

Comme expliqué, dans le pire des cas, cette table peut contenir 30 entrées comme c'est le cas pour les tables de routage. Dès lors, le temps de calcul pour la réception peut atteindre au maximum environ 17 ms soit plus de 2 fois le temps de gestion d'une trame par rapport au réseau de référence. Cela est limitant dans le cas de réseaux denses et pour un routeur proche du 6BR et entraine des pertes de paquets. Enfin, lors de l'émission de trames, indépendamment du type (ICMPv6 ou UDP), le surcoût est d'environ 16%.

Pour l'application, l'utilisation d'Ephemeral implique une baisse du débit. Ainsi, indépendamment du débit radio, pour le réseau de référence, l'application peut espérer traiter en émission environ 103 trames entières en 1s. Lorsqu'Ephemeral est utilisé, ce débit est réduit à 87 trames/s. Ce nouveau débit implique que le réseau ne pourra traiter des applications où il est nécessaire de communiquer toutes les 0,01s. Les applications espérées pour les réseaux de capteurs sont des applications bas débits où le scénario type implique une collecte de l'ordre d'une donnée par seconde. Ce débit reste convenable pour le type d'application envisagée dans les réseaux de capteurs.

Le deuxième point crucial lors du déploiement d'une solution de sécurité dans des nœuds contraints concerne l'empreinte mémoire. Les cartes Openmote possèdent une mémoire RAM de 32 KB et une mémoire ROM de 512 KB. Dans la partie RAM de la mémoire, une zone est dédiée à la pile. La valeur maximale que peut prendre cette dernière est fixée dans Contiki à 256 octets sur les 32 KB disponibles. Nous allons comparer les tailles d'image mémoire pour Ephemeral et l'implémentation de référence pour un même scénario.

Les valeurs obtenues pour un nœud routeur sont données dans la Figure 7.8.

Ephemeral prend 1388 octets soit 14% de mémoire RAM en plus que l'implémentation de référence. La différence est due en grande partie à l'allocation statique de la table de pseudonymes comportant 30 entrées maximum ainsi qu'au matériel de pseudonymes comme la clé ou encore l'IV propre du nœud. En ce qui concerne la taille du code, la ROM passe de 45 092 octets à 55 499 octets soit 10 407 octets supplémentaires dédiés à l'implémentation d'Ephemeral.

Enfin, nous avons voulu quantifier la consommation d'énergie due à Ephemeral. La consommation d'énergie doit être minimale de façon à permettre une durée de vie importante. Ephemeral doit donc apporter un surcoût de consommation limité. Afin de mesurer la consommation d'énergie, une méthode matérielle est utilisée.

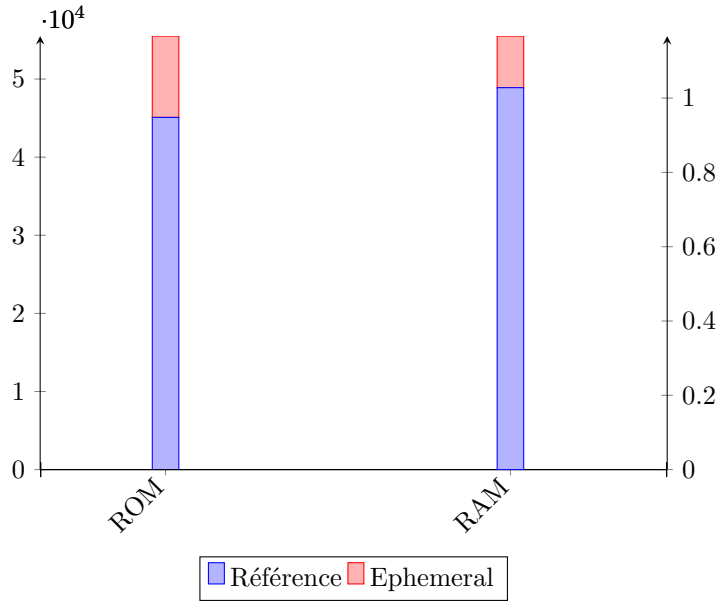


FIGURE 7.8 – Comparaison de l'empreinte mémoire d'Ephemeral.

Une résistance de 1Ω est mise en série entre la source d'énergie et la carte. Cette valeur est choisie de manière à ne pas provoquer de chute de tension trop importante mais également à permettre une conversion automatique de la valeur lue à l'oscilloscope. L'oscilloscope affiche et mesure deux informations. Tout d'abord, la valeur de la tension aux bornes de la résistance de *shunt* est mesurée. Afin d'identifier le début et la fin de la phase d'émission/réception, les GPIO de la carte sont utilisés. Comme pour la mesure du temps de traitement, le GPIO est mis à 1 au début de la phase et remis à 0 à la fin. Le créneau obtenu est superposé avec la valeur de courant mesurée aux bornes de la résistance de *shunt*.

Le réseau est le même que celui utilisé pour le calcul du temps d'émission/réception. Ce réseau est donc composé de notre 6BR, de la TOE et d'un troisième nœud. Douze différentes expériences ont été menées pour mesurer la consommation d'énergie en émission puis en réception pour le même type de trames que précédemment.

La Figure 7.9 montre un exemple des courbes obtenues lors de la mesure de courant avec la méthode matérielle. En vert, la courbe de courant d'Ephemeral pour l'émission d'une trame RPL et en bleu celle pour le réseau de référence.

Les valeurs obtenues pour le réseau de référence correspondent à celles attendues. En effet, d'après la documentation technique de l'Openmote, les valeurs indiquées pour la consommation de courant sont de 24 à 27 mA lorsque la radio est en attente d'un signal et 34 mA pour l'émission. D'après notre courbe, la valeur du courant mesurée avant l'émission, c'est à dire lorsque la radio attend la réception d'un signal est d'environ 26.44 mA. En ce qui concerne l'émission de la trame RPL, le courant mesuré est approximativement de 30.70 mA, ce qui est un peu plus faible que la valeur indiquée mais compte tenu des imprécisions de mesure, reste dans le même ordre de grandeur. Ces résultats proches de ceux donnés permettent de valider la méthodologie utilisée.

En ce qui concerne Ephemeral, les valeurs de courant sont quasi similaires à celles obtenues pour le réseau de référence et ceux pour les 12 expériences menées. Ephemeral ne modifie pas les valeurs de courant en phase de veille et en phase de fonctionnement. Néanmoins, comme nous l'avons vu précédemment et comme le montre la Figure 7.9, Ephemeral augmente le temps d'émission et de réception par rapport au réseau de référence, ce qui influe sur la consommation d'énergie.

Nous allons donc mettre en relation ces deux mesures afin d'estimer cette énergie consommée et le surcoût d'Ephemeral.

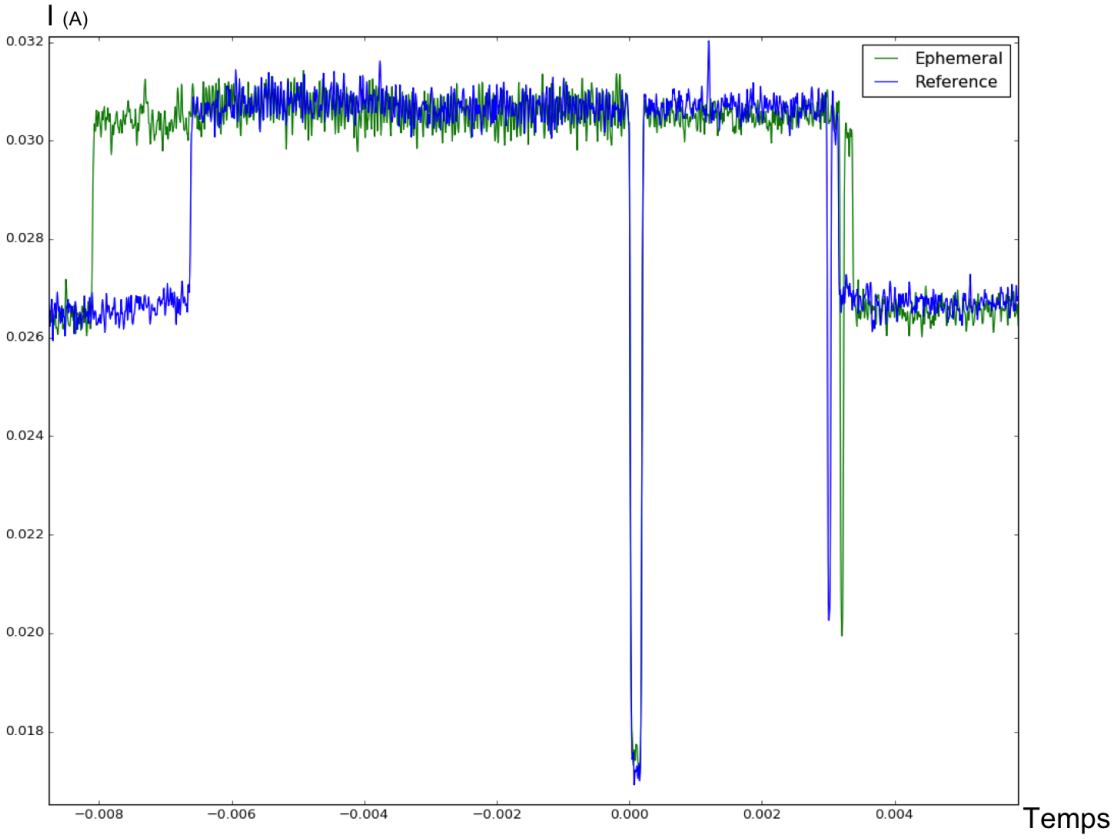


FIGURE 7.9 – Mesure de courant pour l'émission d'une trame RPL.

Type	Réseau	Taille (octets)	Energie (µJ)	augmentation (%)
RX UDP	Référence	86	6.10	23
	Ephemeral	92	7.54	
RX ICMPv6	Référence	102	7.33	35
	Ephemeral	125	9.89	
TX UDP	Référence	86	9.26	17
	Ephemeral	92	10.84	
TX ICMPv6	Référence	81	9.14	16
	Ephemeral	87	10.68	

TABLE 7.3 – Comparaison de l'énergie consommée.

Notre réseau de référence consomme, d'après la Table 7.3, autour de 7 µJ pour la réception et 9 µJ pour l'émission. Ephemeral consomme près de 10 µJ pour la réception d'une trame de 125 octets et autant pour l'émission de trames de taille proche de 90 octets.

D'après les expériences, Ephemeral augmente donc de 23% la consommation d'énergie pour la réception d'une trame d'application et de 17% pour l'émission d'une trame d'application par rapport au réseau de référence.

Le cahier des charges de la solution idéale indique que celle-ci doit limiter le coût nécessaire pour la génération des pseudonymes. Ephemeral utilise un AES logiciel. L'instanciation de l'AES introduit un surcoût de 212 µs pour la génération d'un pseudonyme. Au maximum, la vérification du pseudonyme source reçu introduit un surcoût de 6,6ms.

Un autre critère concerne la diffusion du matériel nécessaire à la dissimulation d'adresses. Celui-ci ne doit pas introduire de trafic ni de coût supplémentaire. Ephemeral n'introduit pas de trafic supplémentaire. Néanmoins, il nécessite l'émission de trames de taille plus importante. Deux paramètres sont donc à prendre en compte. Le temps nécessaire à la création et au traitement de la trame contenant le matériel (trame ICMP) et l'énergie consommée pour ces trames. Pour les trames ICMPv6 étudiées, Ephemeral augmente de 35% le temps de traitement en réception et de 16% le temps de traitement en émission. L'énergie consommée augmente dans les mêmes proportions par rapport à la consommation observée pour la référence. Néanmoins, ces envois de trames ICMPv6 ne sont pas fréquents et restent limités dans le cas d'un réseau statique.

Enfin, pour les communications concernant l'application, Ephemeral offre un surcoût de 14% pour des trames UDP de même taille (en énergie et en temps).

D'autres critères du cahier des charges nécessitent d'observer le réseau lors de différentes phases de vie.

7.3.4 Comportement des nœuds

Dans cette partie, le réseau décrit dans la partie 7.3.1 ainsi que la surveillance ont été utilisés pour analyser le comportement des nœuds implémentant Ephemeral lors de différentes phases. Nous voulons étudier l'impact d'Ephemeral sur le fonctionnement nominal du réseau.

Le cahier des charges indique que la solution idéale ne doit pas nuire aux protocoles déjà déployés. Les protocoles étudiés sont RPL pour le routage et l'envoi de trames d'application via UDP. Lors des expériences, grâce au trafic collecté, on peut voir que les nœuds communiquent leurs données et que celles-ci sont routées jusqu'au nœud *root*. Pour l'application choisie, i.e. pour l'émission toutes les 10s d'une trame UDP à destination d'un autre réseau (communication IR), le surcoût de traitement des trames dû à Ephemeral n'introduit pas de pertes de paquets supplémentaires. Les pertes observées viennent du média et des interférences. Les tables de routage enregistrées par les différents types de nœuds surveillés sont identiques à celles obtenues pour le réseau de référence. On observe le fonctionnement classique de SLAAC basé sur les adresses MAC réelles des nœuds.

Afin de vérifier l'évolution dynamique des tables de routage, nous avons testé l'ajout de nouveau nœud dans un réseau déjà déployé. Nous avons fait des expériences où le nœud est ajouté après 6 minutes de fonctionnement du réseau. L'ajout est réalisé à différents endroits du réseau et proche de différents types de nœuds (routeur ou feuille). Le nœud entrant peut également jouer l'un ou l'autre des rôles. La mobilité des nœuds a été également testée. Les tables de routage observées dans le réseau avec Ephemeral suivent le comportement nominal observé dans le réseau de référence. Les nœuds arrivent à rejoindre le réseau, à s'associer, récupérer les informations nécessaires à la création des pseudonymes de leurs voisins et à communiquer en utilisant les bonnes adresses. Ephemeral est donc compatible avec le déploiement des réseaux de capteurs où les nœuds peuvent se déplacer, disparaitre ou être ajoutés.

Le dernier critère du cahier des charges concerne la flexibilité et l'adaptabilité du protocole de génération des pseudonymes face à des problèmes de désynchronisation. Notre solution ne nécessite pas d'être temporellement synchronisé. Nous avons vu avec l'ajout de nœuds en cours d'expériences, que ceux-ci réussissaient à communiquer sans pertes de performance.

Néanmoins, comme Ephemeral transmet en clair les valeurs des compteurs ceux-ci peuvent être utilisés pour identifier les nœuds intéressants (nœud *root*, routeur proche du nœud *root*...). Un mécanisme est prévu dans l'implémentation d'Ephemeral et doit se mettre en place lorsqu'une trop grande désynchronisation est identifiée. En effet, en réel, les nœuds sont autonomes et un phénomène de *jitter* apparait. Nous avons délibérément modifié le programme de certains nœuds du réseau afin d'introduire des incrémentations de compteurs différentes. Ainsi, nous avons pu tester le comportement du réseau face à cette désynchronisation. Avec l'application choisie, le mécanisme se déclenche dès la première trame échangée avec un compteur désynchronisé. Un attaquant ne pourra donc pas utiliser cette désynchronisation pour identifier les nœuds. Il serait intéressant de tester cette désynchronisation pour d'autres applications où la fréquence d'émission des trames UDP est plus élevée afin de déterminer si le mécanisme a le temps de se déclencher avant une nouvelle émission.

De même, le bon fonctionnement des mécanismes de sûreté ajoutés pour gérer les cas problématiques dûs au partage des IV et à leur régénération ont été testés. Le réseau a été observé lorsque ces mécanismes étaient déclenchés et lorsque ceux-ci n'avaient pas besoin d'être utilisés. Malgré la présence de ces mécanismes, les communications de l'application avaient bien lieu tout comme le trafic pour le maintien des tables de routage. Les mécanismes ne nuisent donc pas au comportement du réseau.

Le portage d'Ephemeral sur les nœuds Openmote montre que le fonctionnement nominal du réseau est préservé.

7.4 Conclusion

Le nouveau schéma de protection de la vie privée, Ephemeral, permet à des nœuds contraints de générer et de gérer des pseudonymes dynamiques à utiliser lors des communications pour dissimuler les adresses MAC. Grâce à la sécurité MAC, les adresses Réseaux sont chiffrées.

Le déploiement réel d'Ephemeral permet de valider son comportement lorsque les nœuds sont mobiles. Il permet d'inclure les compteurs utilisés dans le processus de calcul des pseudonymes dans les en-têtes MAC ce qui permet à un nœud d'être isolé ou désynchronisé sans perturbation des communications.

Grâce à la dissimulation d'adresses, un attaquant possède des difficultés pour collecter les informations de vie privée utiles à la mise en place d'attaques actives ciblées. L'adresse ayant une durée de vie limitée, l'attaquant doit identifier le nœud à attaquer par une écoute passive puis lancer son attaque active avant le changement de pseudonyme. Si le *timing* est correctement choisi, ces étapes sont impossibles à réaliser avec Ephemeral dans le temps imparti.

Néanmoins, le changement d'adresses apporte des pertes de performances que nous avons quantifiées.

Dans cette partie, nous avons tout d'abord voulu positionner Ephemeral vis-à-vis de l'état de l'art et plus particulièrement vis-à-vis de MT6D, solution qui nous a paru être la plus adaptée aux WSN 6LoWPAN. MT6D permet également l'utilisation de pseudonymes dynamiques dans les réseaux de capteurs mais contrairement à Ephemeral nécessite de remplacer définitivement la valeur des adresses MAC et IP par les pseudonymes générés.

Grâce à une implémentation des deux solutions dans Contiki 3.0 ainsi que des tests en simulation, nous avons pu affirmer qu'Ephemeral possède des performances meilleures que MT6D. Contrairement à cette dernière, Ephemeral n'introduit pas de trafic supplémentaire pour le partage et la régénération du matériel de protection de la vie privée. Il est entièrement compatible avec les standards et protocoles des réseaux 6LoWPAN.

Fort de ces résultats, nous avons déployé Ephemeral sur des cartes Openmote afin de quantifier l'impact sur l'empreinte mémoire mais également la consommation d'énergie ou encore le surcoût de calculs. Nous avons observé l'adéquation d'Ephemeral avec le comportement nominal d'un réseau sans solution de protection de la vie privée. Cette étude nous a permis de quantifier le coût de déploiement de notre solution pour le déploiement choisi.

Ce déploiement peut toutefois être amélioré.
Notre solution est indépendante du choix du matériel déployé. Néanmoins, les cartes Openmote ne permettent pas l'utilisation d'un AES *hardware*. Il serait donc intéressant de choisir des cartes disposant d'un AES matériel afin d'étudier les performances d'Ephemeral sur la consommation d'énergie et le temps de calcul lorsque la génération et la vérification des pseudonymes est effectuée par un crypto processeur.

Une autre perspective d'évolution concerne l'amélioration du fonctionnement d'Ephemeral et notamment l'étape de vérification des pseudonymes. Lors de la réception d'une trame contenant des pseudonymes, nous avons vu qu'Ephemeral pouvait apporter des calculs supplémentaires pour vérifier le pseudonyme source de la trame qui nuisaient à la consommation d'énergie mais également aux débits. En effet, dans le *design* prévu actuellement, lorsqu'un nœud reçoit une trame, il va vérifier si le pseudonyme destination correspond à son adresse MAC. Si c'est le cas, il va alors vérifier le pseudonyme source et le comparer aux adresses enregistrées dans sa table de routage. Ne connaissant pas l'IV à utiliser, il va alors réaliser une recherche exhaustive avec tous les IV qu'ils possèdent dans sa table jusqu'à obtenir une adresse MAC réelle connue ou atteindre la fin de la table.

Un attaquant peut utiliser ce comportement non optimal afin de réaliser une exhaustion des ressources. Il aura simplement à forger une trame contenant une adresse MAC source aléatoire mais avec l'adresse destination contenant un pseudonyme valide. Le *payload* peut être forgé aléatoirement. Le nœud récepteur va alors accepter la trame et va réaliser la recherche exhaustive sur l'adresse source sans réussir à retrouver une adresse réelle.

Une contre mesure consiste à mettre en place des mécanismes d'authentification de l'émetteur de la trame. Un mécanisme utilisant un MAC est déjà prévu dans le standard IEEE 802.15.4. Néanmoins, celui-ci est traité après le traitement de l'adresse source et le déchiffrement de la trame. Il faut alors ajouter un nouveau MAC calculé sur les champs de l'en-tête de la couche MAC et vérifié après traitement de l'adresse destination. Ainsi, même si l'attaquant peut réutiliser les pseudonymes, ne connaissant pas la clé, il ne pourra forger un MAC valide. La trame sera rejetée avant le traitement exhaustif de l'adresse source. Néanmoins, cette contre mesure ne permet pas d'améliorer le comportement d'Ephemeral et même en l'absence d'attaquant, la recherche exhaustive doit avoir lieu. Cette méthodologie peut être améliorée en permettant au nœud, par exemple, de retrouver ou reconstruire l'IV ou une partie de l'IV de la source grâce aux différents champs de l'en-tête MAC comme c'est le cas pour la sécurité définie par le standard IEEE 802.15.4.

Néanmoins, dans le cas d'Ephemeral, cela revient à ajouter encore de nouveaux champs dans l'en-tête, les champs disponibles ne permettant pas un tel travail et donc apporter un surcoût sur la taille. La taille ne devant pas excéder 127 octets sous peine de fragmentation et donc de surcoût de trafic, la mise en place de cette solution est compliquée et coûteuse. Il serait donc intéressant de comparer la consommation d'énergie lors de l'utilisation des deux mécanismes.

Partie 8

Conclusion et perspectives

8.1 Contributions de la thèse

Cette thèse aborde la protection de la vie privée dans les réseaux de capteurs IEEE 802.15.4. Les communications sans fil facilitent la mise en place d'écoutes passives et d'analyse de trafic. Les identifiants permanents utiles pour le routage représentent une vulnérabilité menant à des attaques de sécurité plus efficaces et rapides. Nous avions pour objectif de fournir une solution aux problèmes liés à la collecte massive de ces identifiants par écoute passive. Pour cela, nous avons étudié les attaques et les contre mesures de l'état de l'art. Deux grands axes de recherche ont été analysés. Le premier s'intéresse aux solutions proposant l'anonymat. Dans le second, les solutions permettent l'utilisation de pseudonymes à la place des adresses MAC et/ou IP. Nous avons montré les limites de certaines de ces solutions et leur manque de compatibilité avec les protocoles déjà déployés (routage, adressage). Nous avons proposé Ephemeral une solution adaptée aux réseaux IEEE 802.15.4.

Dans un premier temps, nous nous sommes intéressés aux informations accessibles par écoute passive ainsi qu'aux capacités de l'attaquant et au coût nécessaire pour la mise en place de l'attaque. Nous avons déployé deux plateformes de test afin d'identifier les fuites d'informations dans des réseaux sécurisés. Ces plateformes basées sur le standard IEEE 802.15.4 permettaient de tester deux technologies largement utilisées : ZigBee et 6LoWPAN. Nous avons montré que même en déployant des intercepteurs bas prix de faible portée les informations disponibles étaient suffisantes pour apporter de nombreuses vulnérabilités exploitables par un attaquant. Les identifiants collectés avec des capacités ordinaires permettaient de mener des attaques ciblées dommageables pour les WSN.

Nous avons identifié MT6D comme l'une des solutions de l'état de l'art la mieux adaptée aux contraintes des réseaux 6LoWPAN. Elle permet l'utilisation des pseudonymes dynamiques auto générés grâce à une fonction de hachage à utiliser pour les adresses IPv6 et les adresses MAC. Dans cette thèse, plusieurs travaux ont été réalisés sur MT6D. Tout d'abord, une analyse théorique de son principe de fonctionnement a été réalisée. Plusieurs dysfonctionnements et inconvénients ont pu être relevés. Afin de corroborer ces résultats et d'analyser plus en profondeur le comportement du réseau, nous avons adapté le déploiement décrit dans [115] par Preiss, Sherburne et al. à un réseau multi sauts de nœuds embarquant la version 3.0 de Contiki. Nous avons pu comparer les performances et le comportement de MT6D avec un réseau de référence sans solution de protection des adresses pour un cas d'usage identique. MT6D augmente de 49% le nombre total de trames échangées (UDP et ICMPv6) par rapport au réseau de référence pour un même cas d'usage. De par son fonctionnement, MT6D nécessite de mettre à jour les tables de routage ainsi que le DODAG ce qui provoque un surcoût de trames de contrôle et nuit à la qualité de service du système. Ainsi environ 29% des trames UDP ne sont pas émises. De plus, MT6D est sensible aux désynchronisations possibles entre les nœuds du réseau. Lors d'une désynchronisation le nombre de trames RPL échangées pour l'avertissement des nouveaux pseudonymes augmente et donc détériore les performances réseau. Elle nécessite alors une synchronisation des nœuds compliquée à réaliser dans des réseaux denses.

L'analyse théorique de MT6D nous a permis de rédiger les spécifications de notre solution de dissimulation des adresses pour le monde des réseaux sans fil contraints. Six critères ont été retenus.

Nous avons ainsi introduit Ephemeral. Son fonctionnement repose sur le déploiement de la sécurité au niveau de la couche MAC. Ephemeral permet de remplacer les adresses MAC IEEE 802.15.4 par des pseudonymes dynamiques auto générés à l'aide d'un AES en mode compteur. Nous avons déployé Ephemeral dans Contiki 3.0 et avons comparé ses performances en simulation à celles de MT6D et à celles du réseau de référence. Le même réseau que celui utilisé pour évaluer MT6D a été déployé. Ephemeral améliore de 16% la qualité de service par rapport à MT6D. Néanmoins, du fait de l'ajout de calculs supplémentaires, des trames sont perdues. Ephemeral réduit ainsi le nombre de trames UDP d'environ 2,6%. Notre schéma ne nécessite pas de synchroniser les horloges des nœuds pour permettre la vérification des pseudonymes. Néanmoins, les nœuds ont besoin de connaitre le matériel nécessaire aux calculs des différents AES en mode compteur. Ephemeral n'introduit pas de trafic supplémentaire pour le changement de pseudonymes ou pour la diffusion du matériel de protection des adresses mais modifie le format et donc la taille des trames ce qui peut impacter la consommation d'énergie et ne peut être étudié en simulation. Nous avons évalué Ephemeral lors d'un déploiement réel avec une vraie radio afin d'identifier son comportement lorsque des pertes de paquets peuvent apparaitre. Nous avons également évalué l'impact d'Ephemeral sur trois critères importants dans les WSN. Les nœuds étant contraints en mémoire, nous avons évalué l'empreinte mémoire de notre solution vis-à-vis de celle du réseau de référence. Ephemeral apporte un surcoût de 14% de l'utilisation de la RAM et 23% pour la taille du code. Le second critère concerne le débit maximal de gestion des trames. Nous avons quantifié le temps complet nécessaire pour l'émission et la réception d'une trame Ephemeral. La baisse de débit de traitement observée lors de l'utilisation d'Ephemeral par rapport au réseau de référence est de 15% ramenant celui-ci à 87 trames/s. Enfin, nous avons étudié la consommation d'énergie dans le réseau embarquant Ephemeral. Dans le pire des cas, Ephemeral augmente de 35% l'énergie consommée. Ephemeral consomme près de 10 μJ pour la réception d'une trame de MTU quasi maximal et autant pour l'émission de trames de taille proche de 90 octets.

8.2 Perspectives

Lors de cette thèse de nombreux travaux ont été réalisés. Néanmoins, il existe plusieurs perspectives d'évolution ou de continuité.

Nous avons étudié dans la partie 7 l'efficacité de notre schéma Ephemeral dans le cadre d'un déploiement réel d'un réseau 6LoWPAN sur cartes Openmote embarquant l'OS Contiki. Nous avons analysé les performances et lacunes de ce déploiement.
Nous avons ainsi pu remarquer que des pertes de paquets plus importantes existaient lorsque le réseau embarquait Ephemeral. Ce problème est dû aux temps de calcul supplémentaires introduit par Ephemeral qui combiné au manque de gestion des queues de Contiki, ne permettait pas de traiter les trames et d'en recevoir de nouvelles simultanément. Les trames entrantes étaient alors ignorées tant que le traitement de la trame en cours n'était pas terminé. Nous avons donc identifiés les limites de notre solution mais également celles de Contiki. Il serait alors intéressant de tester Ephemeral avec différents OS tels que FreeRTOS ou encore RIOT. Ces différents tests permettraient d'analyser Ephemeral en présence d'un mécanisme de gestion des queues et ainsi d'identifier l'impact provenant uniquement d'Ephemeral sur la perte de paquets dû aux calculs supplémentaires.

Ephemeral est déployé au niveau de la couche MAC. Les données de la couche Physique sont toujours accessibles à un attaquant. Il peut ainsi étudier les métriques physiques comme la puissance du signal ou la qualité du canal afin d'identifier de manière unique les nœuds. Il serait intéressant de voir si lorsqu'Ephemeral est déployé un attaquant est capable grâce aux métriques physiques de relier plusieurs pseudonymes ensemble avec une forte probabilité. Une contre mesure au niveau de la couche Physique pourrait alors être déployée en complément comme la variation de la puissance d'émission. Les performances dans le cas du déploiement des deux contre mesures combinées doivent être analysées afin d'étudier la faisabilité dans ces réseaux contraints.

Ephemeral a été utilisé pour cacher les adresses MAC IEEE 802.15.4. Néanmoins, d'autres standards sont utilisés dans l'IoT comme le WiFi ou le Bluetooth. Nous avons vu dans nos analyses de la vie privée effectuées sur ces deux réseaux que des fuites d'adresses étaient également incluses dans les métadonnées de

ces standards. Une piste d'évolution pourrait concerner la portabilité d'Ephemeral sur ces standards moins contraints mais très utilisés. Plusieurs problèmes seront alors à résoudre. Une analyse des protocoles de routage définis par ces standards et de leur fonctionnement devra être réalisée afin d'étudier la faisabilité d'utiliser ces protocoles pour le partage du matériel de génération des pseudonymes (périodicité, format, fonctionnement...). Le format de l'en-tête MAC devra également être analysé pour identifier l'espace disponible pour inclure les champs concernant les compteurs.

Enfin, le déploiement que nous avons réalisé avait pour but de reproduire un *smart office*. Nous avons déployé 20 nœuds. Néanmoins, aucun cas réel d'utilisation ou d'application n'étaient déployées. Une perspective serait de déployer Ephemeral dans le but d'obtenir une solution robuste pour un cas d'usage spécifique. Un déploiement dans un système domotique pourrait être une application intéressante. Il faudrait alors embarquer Ephemeral sur de nombreux nœuds mais également sur une longue période. Le réseau pourrait être déployé dans un bâtiment et des applications pourraient ainsi utiliser les données collectées. Un retour d'expérience avec de réels utilisateurs pourrait être obtenu.

Bibliographie

[1] Bernd Michael Dorge and Thomas Scheffler. Using ipv6 and 6lowpan for home automation networks. In *Consumer Electronics-Berlin (ICCE-Berlin), 2011 IEEE International Conference on*, pages 44–47. IEEE, 2011.

[2] Yogendra S Dohare, Tanmoy Maity, Partha Sarathi Paul, and Partha S Das. Design of surveillance and safety system for underground coal mines based on low power wsn. In *Signal Propagation and Computer Technology (ICSPCT), 2014 International Conference on*, pages 116–119. IEEE, 2014.

[3] Venkatesh Rajendran, Katia Obraczka, and Jose Joaquin Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless networks*, 12(1) :63–78, 2006.

[4] Jamal N Al-Karaki and Ahmed E Kamal. Routing techniques in wireless sensor networks : a survey. *IEEE wireless communications*, 11(6) :6–28, 2004.

[5] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A smart home is no castle : Privacy vulnerabilities of encrypted iot traffic.

[6] Osama Majeed Butt, Muhammad Zulqarnain, Tallal Majeed Butt, and Abdul Sattar Malik. Development of smart home automation system by using controlled network. *JOURNAL OF FACULTY OF ENGINEERING & TECHNOLOGY*, 23(2) :xx–xx, 2016.

[7] Jia Wang, Asad Khalid Raja, and Zhibo Pang. Prototyping and experimental comparison of ir-uwb based high precision localization technologies. In *Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), 2015 IEEE 12th Intl Conf on*, pages 1187–1192. IEEE, 2015.

[8] IEEE Standards Association et al. 802.15. 6-2012 ieee standards for local and metropolitan area networks–part 15.6 : Wireless body area networks.

[9] SMART CITIES. Trace analysis and mining for smart cities : issues, methods, and applications. *IEEE Communications Magazine*, 121, 2013.

[10] Tobias Kowatsch and Wolfgang Maass. Privacy concerns and acceptance of iot services. *The Internet of Things 2012 : New Horizons*, pages 176–187, 2012.

[11] Henrich C Pöhls, Vangelis Angelakis, Santiago Suppan, Kai Fischer, George Oikonomou, Elias Z Tragos, Rodrigo Diaz Rodriguez, and Theodoros Mouroutis. Rerum : Building a reliable iot upon privacy-and security-enabled smart objects. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2014 IEEE*, pages 122–127. IEEE, 2014.

[12] Jim Bound, Bernie Volz, Charles E Perkins, Ted Lemon, and Mike Carney. Dynamic host configuration protocol for ipv6 (dhcpv6). 2003.

[13] Susan Thomson. Ipv6 stateless address autoconfiguration. 1998.

[14] Deepak Goyal and Malay Ranjan Tripathy. Routing protocols in wireless sensor networks : A survey. In *Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on*, pages 474–480. IEEE, 2012.

[15] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (aodv) routing. Technical report, 2003.

[16] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (olsr). Technical report, 2003.

[17] Linus Wallgren, Shahid Raza, and Thiemo Voigt. Routing attacks and countermeasures in the rpl-based internet of things. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[18] Nettle aes. `https://git.lysator.liu.se/nettle/nettle/tree/master`, consulté le 2017-02-01.

[19] Tiny-aes. `https://github.com/kokke/tiny-AES128-C`, consulté le 2017-02-01.

[20] Laurent Eschenauer and Virgil D Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47. ACM, 2002.

[21] François Delaveau, Andreas Mueller, Christiane Kameni Ngassa, René Guillaume, Renaud Molière, and Gerhard Wunder. Perspectives of physical layer security (physec) for the improvement of the subscriber privacy and communication confidentiality at the air interface. *Perspectives*, 27 :28, 2016.

[22] Christine Hennebert and Jessye Dos Santos. Security protocols and privacy issues into 6lowpan stack : A synthesis. *Internet of Things Journal, IEEE*, 1(5) :384–398, 2014.

[23] Wassim Znaidi. *Quelques propositions de solutions pour la sécurité des réseaux de capteurs sans fil*. PhD thesis, Rapport de thèse, 2010.

[24] Jaydip Sen. Security and privacy challenges in cognitive wireless sensor networks. *Cognitive Radio Technology Applications for Wireless and Mobile Ad hoc Networks*, pages 194–232, 2013.

[25] Jianliang Zheng, Myung J Lee, and Michael Anshel. Toward secure low rate wireless personal area networks. *Mobile Computing, IEEE Transactions on*, 5(10) :1361–1373, 2006.

[26] Syed Muhammad Sajjad and Muhammad Yousaf. Security analysis of ieee 802.15. 4 mac in the context of internet of things (iot). In *Information Assurance and Cyber Security (CIACS), 2014 Conference on*, pages 9–14. IEEE, 2014.

[27] Sunil Ghildiyal, Amit Kumar Mishra, Ashish Gupta, and Neha Garg. Analysis of denial of service (dos) attacks in wireless sensor networks. *IJRET : International Journal of Research in Engineering and Technology*, 3 :2319–1163, 2014.

[28] Wazir Zada Khan, Xiang Yang, Mohammed Y Aalsalem, and Quratulain Arshad. Comprehensive study of selective forwarding attack in wireless sensor networks. *International Journal of Computer Network and Information Security*, 3(1) :1, 2011.

[29] Ioannis Krontiris, Thanassis Giannetsos, and Tassos Dimitriou. Launching a sinkhole attack in wireless sensor networks ; the intruder side. In *Networking and Communications, 2008. WIMOB'08. IEEE International Conference on Wireless and Mobile Computing,*, pages 526–531. IEEE, 2008.

[30] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. The sybil attack in sensor networks : analysis & defenses. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 259–268. ACM, 2004.

[31] Priya Maidamwar and Nekita Chavhan. A survey on security issues to detect wormhole attack in wireless sensor network. *International Journal on AdHoc Networking Systems (IJANS) Vol*, 2 :37–50, 2012.

[32] Wenyuan Xu, Ke Ma, Wade Trappe, and Yanyong Zhang. Jamming sensor networks : attack and defense strategies. *IEEE network*, 20(3) :41–47, 2006.

[33] Hang Liu, Hairuo Ma, Magda El Zarki, and Sanjay Gupta. Error control schemes for networks : An overview. *Mobile networks and Applications*, 2(2) :167–182, 1997.

[34] David D Falconer, Fumiyuki Adachi, and Bjorn Gudmundson. Time division multiple access methods for wireless personal communications. *IEEE Communications Magazine*, 33(1) :50–57, 1995.

[35] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4) :11–25, 2001.

[36] Wang Xin-Sheng, Zhan Yong-zhao, Xiong Shu-ming, and Wang Liang-min. Lightweight defense scheme against selective forwarding attacks in wireless sensor networks. In *Cyber-Enabled Distributed Computing and Knowledge Discovery, 2009. CyberC'09. International Conference on*, pages 226–232. IEEE, 2009.

[37] Chris Karlof and David Wagner. Secure routing in wireless sensor networks : Attacks and countermeasures. *Ad hoc networks*, 1(2) :293–315, 2003.

[38] Luigi Coppolino, Salvatore D'Antonio, Luigi Romano, and Gianluigi Spagnuolo. An intrusion detection system for critical information infrastructures using wireless sensor network technologies. In *Critical Infrastructure (CRIS), 2010 5th International Conference on*, pages 1–8. IEEE, 2010.

[39] Lingxuan Hu and David Evans. Using directional antennas to prevent wormhole attacks. In *NDSS*, 2004.

[40] Tuomas Aura, Pekka Nikander, and Jussipekka Leiwo. Dos-resistant authentication with client puzzles. In *International workshop on security protocols*, pages 170–177. Springer, 2000.

[41] Abror Abduvaliyev, Al-Sakib Khan Pathan, Jianying Zhou, Rodrigo Roman, and Wai-Choong Wong. On the vital areas of intrusion detection systems in wireless sensor networks. *Communications Surveys & Tutorials, IEEE*, 15(3) :1223–1237, 2013.

[42] Syed Obaid Amin, Muhammad Shoaib Siddiqui, Choong Seon Hong, and Jongwon Choe. A novel coding scheme to implement signature based ids in ip based sensor networks. In *Integrated Network Management-Workshops, 2009. IM'09. IFIP/IEEE International Symposium on*, pages 269–274. IEEE, 2009.

[43] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7) :422–426, 1970.

[44] Miao Xie, Song Han, Biming Tian, and Sazia Parvin. Anomaly detection in wireless sensor networks : A survey. *Journal of Network and Computer Applications*, 34(4) :1302–1325, 2011.

[45] Ioannis Krontiris, Tassos Dimitriou, Thanassis Giannetsos, and Marios Mpasoukos. Intrusion detection of sinkhole attacks in wireless sensor networks. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 150–161. Springer, 2007.

[46] Ioannis Krontiris, Thanassis Giannetsos, and Tassos Dimitriou. Lidea : a distributed lightweight intrusion detection architecture for sensor networks. In *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, page 20. ACM, 2008.

[47] Eleni Darra and Sokratis K Katsikas. Attack detection capabilities of intrusion detection systems for wireless sensor networks. In *Information, Intelligence, Systems and Applications (IISA), 2013 Fourth International Conference on*, pages 1–7. IEEE, 2013.

[48] Wireshark. `https://www.wireshark.org/`, consulté le 2016-09-22.

[49] Xi Luo, Xu Ji, and Myong-Soon Park. Location privacy against traffic analysis attacks in wireless sensor networks. In *Information Science and Applications (ICISA), 2010 International Conference on*, pages 1–6. IEEE, 2010.

[50] Jing Deng, Richard Han, and Shivakant Mishra. Countermeasures against traffic analysis attacks in wireless sensor networks. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 113–126. IEEE, 2005.

[51] Alissa Cooper, Fernando Gont, and Dave Thaler. Security and privacy considerations for ipv6 address generation mechanisms. Technical report, 2016.

[52] Walid Saad, Xiangyun Zhou, Merouane Debbah, and H Vincent Poor. Wireless physical layer security : Part 1 [guest editorial]. *IEEE Communications Magazine*, 53(6) :15–15, 2015.

[53] Laurent Bernaille and Renata Teixeira. Early recognition of encrypted applications. In *Passive and Active Network Measurement*, pages 165–175. Springer, 2007.

[54] Björn Muntwyler, Vincent Lenders, Franck Legendre, and Bernhard Plattner. Obfuscating ieee 802.15. 4 communication using secret spreading codes. In *Wireless On-demand Network Systems and Services (WONS), 2012 9th Annual Conference on*, pages 1–8. IEEE, 2012.

[55] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2) :84–90, 1981.

[56] Claudia Diaz and Bart Preneel. Taxonomy of mixes and dummy traffic. In *Information Security Management, Education and Privacy*, pages 217–232. Springer, 2004.

[57] Pandurang Kamat, Wenyuan Xu, Wade Trappe, and Yanyong Zhang. Temporal privacy in wireless sensor networks : Theory and practice. *ACM Transactions on Sensor Networks (TOSN)*, 5(4) :28, 2009.

[58] Wazen M Shbair, Ahmed R Bashandy, and Samir I Shaheen. A new security mechanism to perform traffic anonymity with dummy traffic synthesis. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 1, pages 405–411. IEEE, 2009.

[59] Kiran Mehta, Donggang Liu, and Matthew Wright. Location privacy in sensor networks against a global eavesdropper. In *2007 IEEE International Conference on Network Protocols*, pages 314–323. IEEE, 2007.

[60] Min Shao, Yi Yang, Sencun Zhu, and Guohong Cao. Towards statistically strong source anonymity for sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[61] Ochirkhand Erdene-Ochir, Marine Minier, Fabrice Valois, and Apostolos Kountouris. Enhancing resiliency against routing layer attacks in wireless sensor networks : Gradient-based routing in focus. *International journal on advances in networks and services*, 2011.

[62] Curt Schurgers and Mani B Srivastava. Energy efficient routing in wireless sensor networks. In *Military communications conference, 2001. MILCOM 2001. Communications for network-centric operations : Creating the information force. IEEE*, volume 1, pages 357–361. IEEE, 2001.

[63] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. Anonymity enabling scheme for wireless ad hoc networks. In *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, pages 136–140. IEEE, 2004.

[64] Alireza A Nezhad, Ali Miri, and Dimitris Makrakis. Location privacy and anonymity preserving routing for wireless sensor networks. *Computer Networks*, 52(18) :3433–3452, 2008.

[65] Dijiang Huang. Anonymous certification services. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6. IEEE, 2010.

[66] Ashmita Debnath, Pradheepkumar Singaravelu, and Shekhar Verma. Privacy in wireless sensor networks using ring signature. *Journal of King Saud University-Computer and Information Sciences*, 26(2) :228–236, 2014.

[67] Tao Jiang, Helen J Wang, and Yih-Chun Hu. Preserving location privacy in wireless lans. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 246–257. ACM, 2007.

[68] Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua R Smith. Passive wi-fi : Bringing low power to wi-fi transmissions. In *Symposium on Networked Systems Design and Implementation (NSDI'16)*, page 151, 2016.

[69] Mohammad Fal Sadikin and Marcel Kyas. Rfid-tate : Efficient security and privacy protection for active rfid over ieee 802.15.4. In *Information, Intelligence, Systems and Applications, IISA 2014, The 5th International Conference on*, pages 335–340. IEEE, 2014.

[70] Keiji Takeda. User identification and tracking with online device fingerprints fusion. In *Security Technology (ICCST), 2012 IEEE International Carnahan Conference on*, pages 163–167. IEEE, 2012.

[71] IEEE Standards Association. Part 15.4 : Low-rate wireless personal area networks (lr-wpans), 2003.

[72] IEEE Standards Association. Part 15.4 : Low-rate wireless personal area networks (lr-wpans), 2006.

[73] IEEE Standards Association. Part 15.4 : Low-rate wireless personal area networks (lr-wpans) amendment 1 : Mac sublayer, 2011.

[74] Phillip Rogaway and David Wagner. A critique of ccm. *IACR Cryptology ePrint Archive*, 2003 :70, 2003.

[75] Pierre-Alain Fouque, Gwenaëlle Martinet, Frédéric Valette, and Sébastien Zimmer. On the security of the ccm encryption mode and of a slight variant. In *Applied Cryptography and Network Security*, pages 411–428. Springer, 2008.

[76] Md Hossen, AFM Kabir, Razib Hayat Khan, Abdullah Azfar, et al. Interconnection between 802.15. 4 devices and ipv6 : implications and existing approaches. *arXiv preprint arXiv :1002.1146*, 2010.

[77] Jonathan Hui, David Culler, and Samita Chakrabarti. 6lowpan : Incorporating ieee 802.15. 4 into the ip architecture. *IPSO Alliance White Paper*, 3, 2009.

[78] Stephen E Deering. Internet protocol, version 6 (ipv6) specification. 1998.

[79] R Hinden. Ip version 6 addressing architecture. Technical report, RFC 2373, 1998.

[80] T Winter, P Thubert, T Clausen, J Hui, R Kelsey, P Levis, K Pister, R Struik, and J Vasseur. Rpl : Ipv6 routing protocol for low power and lossy networks, rfc 6550. *IETF ROLL WG, Tech. Rep*, 2012.

[81] Alex Conta and Mukesh Gupta. Internet control message protocol (icmpv6) for the internet protocol version 6 (ipv6) specification. 2006.

[82] Ricardo Silva, Valderi RQ Leithardt, Jorge Sa Silva, Claudio Geyer, Joel Rodrigues, and Fernando Boavida. A comparison of approaches to node and service discovery in 6lowpan wireless sensor networks. In *Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks*, pages 44–49. ACM, 2009.

[83] Uttam Ghosh and Raja Datta. A secure dynamic ip configuration scheme for mobile ad hoc networks. *Ad Hoc Networks*, 9(7) :1327–1342, 2011.

[84] Thomas Narten, Susan Thomson, and Tatuya Jinmei. Ipv6 stateless address autoconfiguration. 2007.

[85] Emilio Ancillotti, Raffaele Bruno, and Marco Conti. On the interplay between rpl and address autoconfiguration protocols in llns. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1275–1282. IEEE, 2013.

[86] Xiaonan Wang, Qi Sun, Yuan Yang, and Dong Wang. Optimal addressing-based routing for 6lowpan. *Computer Standards & Interfaces*, 45 :79–89, 2016.

[87] Pascal Thubert. Objective function zero for the routing protocol for low-power and lossy networks (rpl). 2012.

[88] Philip Levis, T Clausen, Jonathan Hui, Omprakash Gnawali, and J Ko. The trickle algorithm. *Internet Engineering Task Force, RFC6206*, 2011.

[89] Oliver Hahm, Emmanuel Baccelli, Hauke Petersen, and Nicolas Tsiftes. Operating systems for low-end devices in the internet of things : a survey. 2015.

[90] Thomas Watteyne, Vlado Handziski, Xavier Vilajosana, Simon Duquennoy, Oliver Hahm, Emmanuel Baccelli, and Adam Wolisz. Industrial wireless ip-based cyber–physical systems. 2016.

[91] Contiki. `http://www.contiki-os.org/`, consulté le 2016-10-06.

[92] Killerbee. `https://github.com/riverloopsec/killerbee`, consulté le 2017-01-23.

[93] Bjorn Stelte and Gabi Dreo Rodosek. Thwarting attacks on zigbee-removal of the killerbee stinger. In *Network and Service Management (CNSM), 2013 9th International Conference on*, pages 219–226. IEEE, 2013.

[94] Niko Vidgren, Keijo Haataja, Jose Luis Patino-Andres, Juan Jose Ramirez-Sanchis, and Pekka Toivanen. Security threats in zigbee-enabled systems : vulnerability evaluation, practical experiments, countermeasures, and lessons learned. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 5132–5138. IEEE, 2013.

[95] Radosveta Sokullu, Ilker Korkmaz, Orhan Dagdeviren, Anelia Mitseva, and Neeli R Prasad. An investigation on ieee 802.15. 4 mac layer attacks. In *Proc. of WPMC*, 2007.

[96] Douglas J Kelly. *A taxonomy for and analysis of anonymous communications networks*. ProQuest, 2009.

[97] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor : The second-generation onion router. Technical report, DTIC Document, 2004.

[98] Tor project. `https://www.torproject.org/about/overview.html.en`, consulté le 2017-05-09.

[99] Ryan Pries, Wei Yu, Xinwen Fu, and Wei Zhao. A new replay attack against anonymous communication networks. In *2008 IEEE International Conference on Communications*, pages 1578–1582. IEEE, 2008.

[100] Roger Dingledine, Nick Mathewson, and Paul Syverson. Challenges in deploying low-latency anonymity (draft). *Unpublished Manuscript. http ://tor. eff. org/cvs/tor/doc/design-paper/challenges. pdf*, 2005.

[101] Alfredo Matos, Susana Sargento, and Rui L Aguiar. Waypoint routing : A network layer privacy framework. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6. IEEE, 2011.

[102] Sangho Park, Jihyun Bang, Mirim Ahn, Woomin Lee, and Taekyoung Kwon. A method for hiding link layer addresses using bloom filter in wireless sensor networks. *Journal of Internet Services and Information Security (JISIS)*, 4(4) :71–81, 2014.

[103] Xiaonan Wang and Yi Mu. Addressing and privacy support for 6lowpan. *Sensors Journal, IEEE*, 15(9) :5193–5201, 2015.

[104] Nouha Oualha, Alexis Olivereau, and Aymen Boudguiga. Pseudonymous communications in secure industrial wireless sensor networks. In *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*, pages 98–102. IEEE, 2013.

[105] Thomas Narten, Richard Draves, and Suresh Krishnan. Privacy extensions for stateless address autoconfiguration in ipv6. 2007.

[106] Ronald Rivest. The md5 message-digest algorithm. 1992.

[107] Jari Arkko, James Kempf, Brian Zill, and Pekka Nikander. Secure neighbor discovery (send). Technical report, 2005.

[108] Tuomas Aura. Cryptographically generated addresses (cga). 2005.

[109] Tony Cheneau and Maryline Laurent. Using send signature algorithm agility and multiple-key cga to secure proxy neighbor discovery and anycast addressing. In *Network and Information Systems Security (SAR-SSI), 2011 Conference on*, pages 1–7. IEEE, 2011.

[110] Hosnieh Rafiee and Christoph Meinel. Ssas : A simple secure addressing scheme for ipv6 autoconfiguration. In *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*, pages 275–282. IEEE, 2013.

[111] Kristin Lauter. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless communications*, 11(1) :62–67, 2004.

[112] I Tunaru, B Denis, and B Uguen. Location-based pseudonyms for identity reinforcement in wireless ad hoc networks. In *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, pages 1–5. IEEE, 2015.

[113] Fernando Gont. A method for generating semantically opaque interface identifiers with ipv6 stateless address autoconfiguration (slaac). 2014.

[114] Matthew Dunlop, Stephen Groat, William Urbanski, Randy Marchany, and Joseph Tront. Mt6d : A moving target ipv6 defense. In *Military Communications Conference, 2011-Milcom 2011*, pages 1321–1326. IEEE, 2011.

[115] Tanner Preiss, Matthew Sherburne, Randy Marchany, and Joseph Tront. Implementing dynamic address changes in contikios. In *Information Society (i-Society), 2014 International Conference on*, pages 222–227. IEEE, 2014.

[116] Matthew Gilbert Sherburne. *Micro-Moving Target IPv6 Defense for 6LoWPAN and the Internet of Things*. PhD thesis, Virginia Polytechnic Institute and State University, 2015.

[117] Erik Zenner. Nonce Generators and the Nonce Reset Problem. In *Information Security, 12th International Conference, ISC 2009*, volume 5735 of *Lecture Notes in Computer Science*, pages 411–426, Pisa, Italy, September 2009. Springer.

[118] Lander Casado and Philippas Tsigas. Contikisec : A secure network layer for wireless sensor networks under the contiki operating system. In *Nordic Conference on Secure IT Systems*, pages 133–147. Springer, 2009.

[119] M. Ergen. *Chapter 8 IEEE 802.11 Protocols*. 2002.

[120] Bram Bonne, Arno Barzan, Peter Quax, and Wim Lamotte. Wifipi : Involuntary tracking of visitors at mass events. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–6. IEEE, 2013.

[121] Mathieu Cunche, Mohamed-Ali Kaafar, and Roksana Boreli. Linking wireless devices using information contained in wi-fi probe requests. *Pervasive and Mobile Computing*, 11 :56–69, 2014.

[122] Bluetooth contributors. Bluetooth specification version 4.1, 2013.

[123] Amara Korba Abdelaziz, Mehdi Nafaa, and Ghanemi Salim. Survey of routing attacks and countermeasures in mobile ad hoc networks. In *Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on*, pages 693–698. IEEE, 2013.

Annexe A

Le WiFi

L'article 1 du décret n°2009-697 du 16 juin 2009 donne la définition d'une normalisation : "La normalisation est une activité d'intérêt général qui a pour objet de fournir des documents de référence élaborés de manière consensuelle par toutes les parties intéressées, portant sur des règles, des caractéristiques, des recommandations ou des exemples de bonnes pratiques, relatives à des produits, à des services, à des méthodes, à des processus ou à des organisations. Elle vise à encourager le développement économique et l'innovation tout en prenant en compte des objectifs de développement durable."

Deux grands organismes assurent aujourd'hui, au niveau mondial, cet objectif de standardisation : l'IEEE fourni les standards pour les couches basses du modèle OSI et l'IETF pour les couches hautes. C'est notamment grâce à un travail de standardisation que l'internet et la téléphonie sont aujourd'hui matures et utilisés par le grand public.

L'IEEE 802.11 décrit les réseaux locaux sans fil (Wireless Local Area Network (WLAN)) dont le WiFi [119]. Le WiFi a connu beaucoup de changements et d'évolutions depuis sa création.

Le WiFi opère sur différentes bandes de communications. Sa portée est de plusieurs dizaines de mètres en intérieur et plusieurs centaines en extérieur. La couche Physique définit la façon dont les ondes radios vont être modulées. Elle définit le codage canal utilisé pour transmettre les données. Afin de réduire les interférences possibles, la couche Physique utilise l'étalement de spectre. Cette technique initialement utilisée dans le domaine militaire permet d'étaler en fréquence le signal temporel. Le signal est transmis dans une largeur de bande plus grande que celle de départ. Une séquence pseudo aléatoire connue de l'émetteur et du récepteur est alors utilisée.

Au-dessus de la couche Physique, la couche Liaison MAC permet de définir l'interface entre le bus de la machine et la couche Physique. Comme expliqué dans la partie 2, l'en-tête MAC est le seul qui ne peut être caché par l'utilisation du chiffrement lors des communications sans fil. Dans cet en-tête, le standard permet de renseigner jusqu'à 4 adresses de 6 octets chacune. Deux autres champs permettent d'indiquer le sens de la communication (AP vers client ou inverse). Cette information permet à un attaquant passif de reconstruire les communications et donc la topologie. Enfin, un champ "Power Management" indique si la station a la possibilité de se mettre en mode économie d'énergie. Cette information est importante pour mener une attaque par exhaustion de batterie.

Connaitre le format des trames ne suffit pas pour analyser la sécurité et la protection de la vie privée d'un réseau. Il faut également connaitre les protocoles utilisés et notamment ceux assurant la sécurité.

Chaque WLAN possède un SSID (Service Set IDentifier) qui permet de le différencier d'un autre WLAN. Tous les participants souhaitant communiquer au sein du même WLAN doivent partager le même SSID. Le SSID est transmis en clair avec chaque paquet. Il permet aux stations de filtrer les trames MAC venant d'autres BSS. Le BSSID (Basic SSID), quant à lui, identifie un AP. Il correspond à l'adresse MAC de l'AP.

Dans le standard IEEE 802.11, le service de découverte autorise une station WiFi à détecter d'autres stations ou AP. Un moyen de découvrir ces AP consiste à faire du *scanning*.
Il existe deux types de scan : actif ou passif.

Lors du scan passif, l'AP envoie périodiquement son BSSID et seul l'appareil choisi si oui ou non il souhaite se connecter à cet AP et donc d'envoyer à son tour son SSID. Cette technique n'est que très peu implémentée dans les appareils WiFi car elle est très longue et induit donc beaucoup de retard.

Lors du scan actif, c'est l'appareil lui-même qui envoie un signal périodiquement à tous les AP proches afin de connaitre leur BSSID. L'avantage est que cela réduit les temps d'attente et rend donc l'association à un réseau rapide. Malheureusement, si l'appareil s'est déjà connecté à un AP, il va lui envoyer directement son adresse MAC alors qu'il ne va pas forcément se connecter à lui. On peut donc suivre le trajet de l'appareil si celui-ci se déplace. La trame envoyée par les appareils est définie dans le standard IEEE 802.11 et inclut le SSID de l'AP mais également l'adresse MAC de l'appareil. De plus, les demandes sont envoyées jusqu'à ce qu'il y ait association avec un AP.

Pour protéger les communications, de la sécurité a été implémentée dans le standard. Afin d'assurer la sécurité du réseau sans fil WiFi, il est nécessaire de le protéger des intrusions grâce à l'authentification de l'appareil. Il existe deux types d'authentification : *Open system Authentication* ou *Shared Key Authentication*.

La première *Open system Authentication* autorise n'importe quel appareil à se connecter au réseau. C'est donc équivalent à ne pas mettre d'authentification. Dans *Shared Key* seules les stations qui ont la même clé secrète sont autorisées à s'authentifier.

La deuxième protection concerne l'intégrité et la confidentialité des données. Pour chiffrer les données, il est possible d'utiliser une clé WPA ou une clé WEP.

Dans le cas où la clé est de type WEP (*Wired Equivalent Privacy*), l'algorithme de chiffrement à flot est le Rivest Cipher (RC4). Des études ont montrées que ce système de clé WEP n'était pas fiable et que casser la clé était rapide.

Le WPA a donc été introduit afin de rendre plus robuste le chiffrement. Le WPA améliore le niveau de protection des données et le contrôle d'accès. La gestion de la clé est plus robuste. Il permet d'authentifier les données d'origine et de protéger l'intégrité des données.

Le WPA utilise le Temporal Key Integrity Protocol (TKIP) pour la gestion des clés ainsi qu'un MIC pour assurer l'intégrité des données transmises. L'algorithme de chiffrement est l'AES. Au niveau de la couche transport, TLS a été implémenté afin d'assurer la sécurité. Pour l'authentification, l'EAP-TLS (TLS over Extensible Authentication Protocol) a également été utilisé. L'EAP-TLS utilise le PKI qui est basé sur une paire asymétrique de clé publique et clé privée. Les données sont chiffrées avec la clé publique du destinataire des données et ne peuvent être déchiffrées uniquement par la clé privée associée et vice versa.

De plus, une signature digitale peut être ajoutée au message. Pour cela, l'émetteur du message calcule le *hash* de ses données et génère ainsi un Message Digest (MD). Il utilise ensuite sa clé privée pour chiffrer le MD et produire sa signature digitale. On ajoute cette signature au message avant de l'envoyer.

Les protocoles de sécurité et de découverte étudiés, le standard connu, il est maintenant possible de regarder quelles informations de vie privée sont accessibles notamment lors d'une attaque par écoute passive ainsi que les contre mesures de la littérature.

De nombreux articles ont été écrits sur l'utilisation du WiFi et de l'écoute passive. L'importance du nombre d'articles tient au fait que ce standard est encore très utilisé par le grand public et qu'il est donc commun de trouver des appareils utilisant du WiFi.

Deux types d'exploitation sont réalisés grâce aux métadonnées contenues dans l'en-tête MAC : utiliser les informations de l'appareil pour déduire des informations sur le propriétaire et la localisation/suivi et l'identification de l'appareil. Les fuites de vie privée concernant les identifiants ont été très étudiées. Mettre en place une solution de protection de l'identité de l'appareil permet de protéger la vie privée de son utilisateur.

Lors d'un scan actif, l'appareil envoie des trames contenant le SSID des AP auxquels il s'est déjà connecté sans que l'AP ciblé n'ait à le lui demander. Ce scan bien qu'utile pour la qualité de service, offre de grandes fuites d'identités mais également un historique des connexions de notre appareil.

De plus, les adresses MAC source et destination sont visibles facilement lors de l'utilisation du WiFi. En effet, chaque communication inclut dans l'en-tête MAC ces adresses. Enfin, le SSID de la station donne une

information intéressante à un attaquant sur l'appareil. En effet, grâce à lui, on peut déduire la marque de l'appareil connecté.

Ces métadonnées permettent d'identifier les deux parties de la communication. Les fuites d'identités sont donc importantes avec le WiFi d'autant plus que c'est un standard très déployé.

Plusieurs exploitations de ces métadonnées peuvent être réalisées.

Grâce aux autres données collectées, il est possible de faire du suivi des appareils. En effet, Bonne, Barzan et al. dans [120] expliquent comment à l'aide de plusieurs détecteurs (AP) placés judicieusement, il est possible de suivre le mouvement d'un appareil identifié grâce à son adresse MAC. On regarde alors quels détecteurs ont reçus l'adresse MAC en question et quand puis on envoie ces données vers un serveur qui va gérer en temps réel les informations fournies. Ainsi, on peut géolocaliser l'appareil mais également suivre ses déplacements dans le temps. On a donc accès aux endroits où il est allé en regardant les SSID des AP et en utilisant le site wigle.com mais également quand il y était grâce à l'horodatage.

La première contre mesure et la plus simple consiste à mettre son WiFi en mode *off* lorsque celui-ci n'est pas utilisé. Cette solution est contraignante pour l'utilisateur et ne résout pas le problème lors de l'utilisation d'un service WiFi. La deuxième solution proposée dans [120] est de cacher l'adresse MAC de l'appareil en lui substituant un nombre aléatoire.

Cunche, Kaafar et al. dans [121] proposent également une solution d'obfuscation cryptographique. Pour cela, ils utilisent des services de découverte assurant la protection de la vie privée de l'appareil grâce à la cryptographie. Une relation de confiance est instaurée entre les AP et les appareils. De plus, des identifiants chiffrés sont utilisés. Néanmoins, cette solution nécessite des modifications des programmes dans les AP. Or, le nombre important d'AP déployé rend cette tâche compliquée et onéreuse. Enfin, ils proposent de pallier au problème en utilisant une demande de connexion aveugle c'est à dire sans divulgation du SSID. Néanmoins, cette solution ne règle pas le problème de la localisation et du suivi.

Pour conclure, toutes les solutions décrites précédemment et notamment celles concernant l'utilisation de pseudonymes à la place des adresses pourraient être mises en place mais le majeur problème reste qu'il faut modifier le logiciel des AP et au vu du nombre d'AP déployés cela serait un travail inconsidérable. Une solution proposée dans [121] moins compliquée à mettre en œuvre serait une géolocalisation assistée de découverte de service actif qui demande seulement le changement de logiciel des appareils et non des AP. Un AP ne peut être associé qu'à une zone fixe d'environ 100 m. L'appareil n'envoie alors des demandes qu'aux AP de sa zone.

Annexe B

Le Bluetooth

Le Bluetooth [122] est décrit par le standard IEEE 802.15.1. Il provient d'un consortium entre plusieurs industriels. La portée du Bluetooth est de 10 cm à 30 m environ. Cette portée est faible comparé à celle du WiFi, son débit est aussi faible mais ce standard permet une consommation d'énergie très basse. Ce standard est beaucoup utilisé car il est très bon marché et peu encombrant.

La couche Physique s'occupe de l'émission et de la réception des ondes radios. Un réseau Bluetooth possède une topologie en piconet c'est à dire que pour une conversation donnée, un des appareils est considéré comme le maître et tous les autres sont les esclaves. C'est le maitre qui gère les considérations de la couche Physique. Les appareils Bluetooth utilisent l'étalement en fréquence du signal comme le WiFi.

La couche MAC de la norme Bluetooth définit les adresses matérielles des périphériques (BD_ADDR). Ce sont les équivalents des adresses MAC. Leur format est définit dans la Figure B.1.

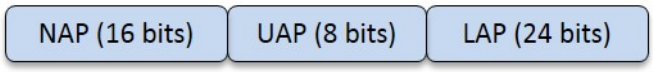


Figure B.1 – Adresse BD_ADDR.

L'adresse Bluetooth est donnée sur 6 octets. Les 2 premiers octets (MSB) forment le Non-significant Address Part (NAP). Ils sont assignés par l'IEEE et publics. On peut consulter la liste des NAP sur le site de l'IEEE. Ces octets dépendent du fabriquant de l'appareil Bluetooth. La marque possède plusieurs plages d'adresses à assigner. L'Upper Address Part (UAP) est composée d'un octet. Il est également assigné par l'IEEE et public. Enfin les 3 derniers octets du Lower Address Part (LAP) sont transmis dans chaque en-tête. 64 adresses sont réservées et ne peuvent être utilisées.

Le standard définit deux types de Bluetooth :

- Basic Rate (BR) avec une option Enhanced Data Rate (EDR)
- Low Energy (LE)

La version Bluetooth LE (appelée BLE par la suite) est poussée comme standard pour l'IoT. Néanmoins, pour l'instant ce standard n'est pas très déployé et beaucoup d'appareils fonctionnent encore avec le Bluetooth "classique".

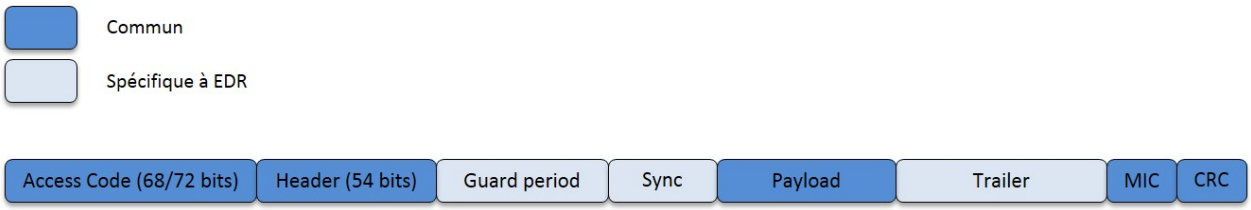


FIGURE B.2 – Format des trames BR et EDR.

Dans la Figure B.2, les champs en bleus foncés sont les champs communs à BR et EDR. Les bleus pales ne sont présents uniquement avec l'option EDR.

Le champ "Access Code" permet de faire de la synchronisation et de l'identification. Il permet également le filtrage d'adresse. Il existe 3 types d'"access code". Le Channel Access Code (CAC) qui permet l'identification du piconet. Il est généré à partir de l'adresse du maitre. Le Device Access Code (DAC) utilisé dans la phase de *paging* et généré à partir de l'adresse esclave. Et le Inquiry Access Code (IAC) utilisé dans la phase de découverte.

Le champ "Header" offre un champ de 3 bits "LT_ADDR" afin d'identifier chaque esclave. La valeur 000 est réservée pour les trames *broadcast*. Le maitre quant à lui ne possède pas de "LT_ADDR". Dès qu'un esclave souhaite se connecter, il envoie au maitre son "AM_ADDR" qui représente son adresse logique de transport. Le maitre utilise alors cet "AM_ADDR" afin de lui faire parvenir sa "LT_ADDR". Si un attaquant assiste à cette phase il peut identifier qui est le maitre et qui sont les esclaves.

Le champ "Payload" de la Figure B.2 est composé de trois parties. La première donne des indications sur la taille du champ suivant, le canal utilisé. La deuxième contient des informations sur l'utilisateur hôte et permet également de router la trame.

L'en-tête MAC Bluetooth offre de nombreux champs comportant des adresses qui pourront être utilisés par la suite pour des attaques sur le nœud. Le Bluetooth définit également des couches hautes. Nous ne les détaillerons pas ici.

Nous allons maintenant voir les protocoles existants dans le Bluetooth afin d'identifier lesquels pourraient être susceptibles de représenter des fuites d'identités.

Le premier protocole utile est la découverte des voisins appelée phase d'*Inquiry*. Cette étape est nécessaire afin d'établir une connexion. Deux acteurs prennent part à cette phase : l'unité qui découvre (esclave) et l'unité que l'on souhaite découvrir (maitre). Lors de ce scan, le maitre attend la réception d'un IAC afin de fournir à l'appareil souhaitant rejoindre le réseau une adresse et un temps d'horloge. L'esclave va, quant à lui, envoyer des paquets ID courts ne comprenant que le champ "Access Code". Il va ensuite envoyer l'IAC sur la bonne fréquence. Il n'envoie aucune information le concernant. Le scan peut durer 10s maximum.

Cette phase effectuée, la phase de *paging* peut commencer. Dans cette phase, l'esclave va tenter de se connecter au réseau qu'il a découvert. Il va écouter sur chaque canal en attente de réception du DAC. Le maitre enverra alors un message ID contenant le DAC de l'unité esclave cherchant à se connecter.

A la fin de cette phase, les deux sont alors connectés et peuvent communiquer.

Etudions maintenant la sécurité permise via le Bluetooth.

Il existe 2 modes afin d'assurer la sécurisation de la couche Liaison mais également l'authentification des unités et la confidentialité. Premièrement, elle peut être activée au niveau service et nécessite un *security manager* qui gère les accès. Elle nécessite une authentification et une autorisation. Enfin, il est possible de mettre la sécurité au niveau Liaison.

La mise en place de la sécurité passe par plusieurs étapes :

- Le *pairing*. Cette phase permet la création des clés. Pour cela, les appareils utilisent SAFER+ (P256 ECDH). Il protège contre l'écoute passive et l'attaque MITM. Chaque appareil génère une paire de clé publique et clé privée à l'aide des courbes elliptiques Diffie Hellman d'ECDH. Le *pairing* est initié par l'émetteur de la clé qui envoie sa clé publique au récepteur qui lui envoie à son tour sa clé publique.
- Le *bonding*. Dans cette phase, la clé est échangée et stockée.
- Les appareils sont ensuite authentifiés. On vérifie que les deux appareils ont la même clé créée par SAFER+ (HMAC-SHA 256).
- Chiffrement (optionnel) : On utilise pour cela l'algorithme AES-CCM. On génère une clé de 8 à 128 bits qui est différente de celle calculée précédemment pour l'authentification. La différence de taille permet de répondre aux différents besoins des pays en termes d'algorithme de cryptographie et facilite une future amélioration du système. Les champs "access code" et "Header" de la Figure B.2 ne sont jamais chiffrés.
- On calcule ensuite un MIC.

Par écoute passive, de nombreuses métadonnées sont disponibles. Grâce à elles, des informations sur l'identité des nœuds sont facilement accessibles. Notamment, les phases de découverte et d'association permettent d'obtenir de nombreuses informations sur les appareils.

Néanmoins, ces phases nécessitent une action synchronisée de chaque coté. De plus, la portée du Bluetooth est limitée. Il est donc nécessaire d'être présent au moment de l'association. Une contre mesure efficace consiste à ne pas réaliser cette association dans un lieu public.

Dans cette thèse nous nous sommes intéressés au Bluetooth dit "classique". Néanmoins, le BLE est poussé pour devenir un standard de l'IoT. Ils seraient donc intéressants de voir comment le standard gère les phases d'association, et si le fonctionnement est identique au Bluetooth "classique", comment le standard prévoit son utilisation dans des réseaux vastes, décentralisés et auto-organisé (i.e. sans intervention humaine) tels que les WSN.

Annexe C

Implémentation de MT6D dans Contiki

C.1 Implémentation dans Contiki 3.0

L'implémentation décrite dans [115] par Preiss, Sherburne et al. pour la version 2.7 de Contiki a été adaptée afin de fonctionner dans Contiki 3.0. Nous avons utilisé l'environnement de simulation décrit dans 7.1.1. Le cas d'usage consiste en un 6BR sans solution de dissimulation et un nœud embarquant MT6D. Nous avons voulu étendre ce cas d'usage à un réseau meshé où tous les nœuds (6BR compris) embarquent MT6D. Plusieurs étapes ont été réalisées.

Dans la première étape, nous avons appliqué MT6D uniquement à l'adresse source du nœud. Le 6BR fonctionne sans pseudonyme. Dans MT6D, les nœuds exécutent, une première fois, SLAAC de manière classique en s'appuyant sur le préfixe IPv6 et leur adresse MAC. Une fois l'association terminée et les tables de routage créées, ces derniers peuvent communiquer dans le réseau et commencer à utiliser les pseudonymes.

Une fenêtre t_i de 10s est déclenchée afin d'assurer la synchronisation des nœuds. Dans la boucle principale du programme, le nœud teste la fin du timer.

Chaque t_i, le nœud appelle une fonction afin de reproduire le même comportement d'initialisation que lors d'une première association. Cette fonction de *setup* définie dans Contiki et appelée lors du démarrage de la carte, permet de convertir l'adresse MAC IEEE 802.15.4 déduite du matériel en adresse 64 bits EUI. Elle permet également d'initialiser les structures pour les adresses IPv6. En réutilisant certaines fonctions du *setup* pour la création des pseudonymes, MT6D s'assure que toutes les structures relatives aux adresses ont bien été mises à jour avec les nouveaux pseudonymes.

Dans la fonction MT6D, une première étape consiste à convertir l'adresse source de manière à la rendre conforme au format Ethernet. Puis, chaque nœud doit mettre à jour ses adresses IPv6. Pour cela, MT6D supprime les anciennes adresses globale et locale du nœud de la structure permettant de les stocker dans la mémoire. Les adresses sont définies comme adresses préférées et possèdent une durée de vie infinie.

A cette étape, le nœud possède donc deux nouvelles adresses IPv6. Il peut maintenant utiliser celles-ci pour mettre à jour son adresse MAC grâce aux mêmes fonctions que celles utilisées dans le *setup*.

Une étape est ensuite nécessaire afin de lier les paramètres radio avec les adresses. Cette étape est utilisée pour filtrer, au niveau matériel, les trames qui ne sont pas destinées au nœud. Une autre solution est de mettre la radio en mode promiscuité. Pour le déploiement de MT6D, le filtrage d'adresses est conservé.

Toutes les adresses et les structures sont donc maintenant mises à jour avec le pseudonyme. Le DODAG peut donc être reconstruit. Cette étape clone le comportement d'un nouveau nœud souhaitant rejoindre le réseau. Le nœud va émettre un DIO. Après une attente de 0.3s, il va émettre un DAO.

Pour émettre un DAO, il est nécessaire, contrairement au DIO de fournir une adresse *unicast* destination. Il faut donc appeler la fonction en lui indiquant cette adresse. Preiss, Sherburne et al. dans [115] proposent le cas d'usage consistant en un nœud communiquant directement avec le 6BR n'utilisant pas de pseudonymes.

Les auteurs ont donc pu indiquer directement l'adresse réelle de celui-ci comme adresse destination du DAO. En revanche, dans un réseau avec plusieurs nœuds utilisant des pseudonymes, le choix de cette adresse doit pouvoir être réalisé à partir de la topologie déployée et non pré indiquée. En effet, celle-ci va être modifiée à chaque changement de pseudonyme. Il a donc fallut faire un choix d'implémentation pour cette adresse destination. Les DAO émis lors du protocole RPL défini par le standard utilisent l'adresse du parent préféré comme adresse destination. Nous avons donc choisis de réutiliser ce comportement pour l'envoi de notre DAO.

Après cette phase de contrôle consistant à construire l'arbre de routage DODAG basé sur les pseudonymes générés, il s'en suit la phase de communication.

Dans ce réseau, seul un 6BR classique et un nœud MT6D communiquent. Afin de tester le comportement de RPL et le bon fonctionnement du routage en présence de pseudonymes dynamiques, un réseau multi sauts doit être mis en place. Notre deuxième contribution a donc été de permettre l'utilisation de MT6D dans un WSN.

C.2 Choix d'implémentation pour le déploiement de MT6D dans un réseau multi sauts

Avant de déployer le réseau multi sauts, nous avons étudié le comportement des pseudonymes dynamiques dans une topologie étoile. Tous les nœuds du WSN à l'exception du 6BR implémentent MT6D.

Le réseau multi sauts de la Figure 7.2 a ensuite été déployé. Dans cette configuration, quand le premier changement de pseudonyme apparait, les communications s'arrêtent.

Plusieurs constats ont pu être dressés par l'analyse du fichier mémorisant les trames brutes envoyées au niveau du 6BR. La trame DIO programmée après la création des pseudonymes est émise correctement. Le nouveau pseudonyme associé est bien ajouté à la table des voisins du récepteur du DIO. En revanche, la trame DAO est émise mais n'atteint jamais l'adresse destination du prochain saut. La table de routage n'est alors jamais mise à jour avec les nouvelles adresses.

Lors d'une association classique, un nœud choisi l'émetteur du premier DIO reçu comme parent préféré. L'OF de RPL défini néanmoins un protocole pour changer de parent préféré. RPL étant un protocole conservateur, il essaie de garder un maximum de chemins déjà établis afin de réduire la consommation inutile. C'est pourquoi, quand un nouveau nœud apparait, il ne peut prétendre devenir le parent préféré d'un nœud déjà associé que s'il revendique une métrique radio meilleure que celle de son parent actuel. En d'autre terme, seul un nœud possédant un rang plus faible pourra prétendre prendre la place du parent préféré. Ce problème est un problème connu de RPL. Dans la fonction de création des pseudonymes de MT6D, nous avons opté pour un comportement similaire à celui de RPL. Nous avons donc choisi l'adresse du destinataire du DAO comme l'adresse du parent préféré. Or, lors de la reconstruction de DODAG quand un nœud reçoit une trame DIO envoyée par un nouveau pseudonyme, il possède déjà une adresse enregistrée comme parent préféré. Cette adresse correspond à l'adresse réelle du parent préféré utilisée pendant la première association. Le réseau déployé est un réseau simulé. La couche radio est parfaite et ses métriques sont indisponibles. Contiki simule alors une valeur équivalente pour tous les nœuds en mode natif. Dans un réseau statique, le parent préféré n'est donc jamais mis à jour. Quand les pseudonymes sont changés, l'adresse de ce parent préféré n'est donc plus valable et le processus de routage est cassé. Ce problème vient de notre environnement de simulation. Afin de remédier à cela et de permettre l'utilisation du simulateur pour analyser les comportements des différentes contre mesures de dissimulation des identifiants, différentes solutions ont été testées.
Tout d'abord, nous avons tenté de changer l'adresse destination du DAO. En effet, dans l'implémentation de [115], l'adresse destination est l'adresse réelle du 6BR et lors du déploiement du réseau en topologie étoile, il n'y a pas eu de problèmes. Un chemin est bien ajouté à la table de routage du 6BR quand il reçoit un DAO.
La table du 6BR grossie à chaque nouveau pseudonyme créé. Nous avons donc tenté d'envoyer un DAO à chaque parent et pas seulement au parent préféré. Nous avons observé que les DAO étaient bien reçus par les nœuds *one-hop* et que leur table de routage est bien mise à jour. Il existe maintenant bien une route pour atteindre le nœud pour des communications du père vers le fils. Néanmoins, comme le mode par défaut de

Contiki est le mode *storing*, le DAO doit être transféré jusqu'au nœud root. Par exemple, dans la Figure 7.2, si le nœud "C" implémente notre nouvelle approche, le DAO qu'il émettra avec son nouveau pseudonyme sera bien reçu par le nœud "B". "B" va alors ajouter une route pour atteindre "C" via son nouveau pseudonyme. Dans un fonctionnement en mode *storing* de RPL, ce DAO doit ensuite être envoyé à "A" pour que celui-ci ajoute une route indiquant que pour atteindre "C" il faut passer par "B". Lors de ce processus, comme le DAO ne concerne pas le nœud lui-même mais un de ses enfants, RPL utilise le parent préféré comme adresse du prochain saut. Dans ce cas, la solution ne marche plus et le DAO n'est pas transmis jusqu'au 6BR. En effet, avec cette solution, "B" garde toujours l'ancienne adresse de "A" comme adresse de son parent préféré. Or "A" ne s'identifie plus avec cette adresse et va donc filtrer la trame. De même, lors de communications du fils vers le père (par exemple, si "B" veut communiquer avec "D"), le nœud ne connait pas la route et va utiliser la route par défaut pour atteindre la destination et va donc envoyer à son parent préféré. Cette solution présente l'inconvénient d'introduire un surcoût de trames DAO proportionnel au nombre de parents de chaque nœud.
La deuxième approche envisagée consiste à modifier les valeurs par défaut de RPL ou d'utiliser des fonctions propres à ce processus. Malheureusement, les solutions utilisées permettent la mise à jour du DODAG mais pas du parent préféré. Nous avons également essayé de procéder à une réparation locale, protocole prévu par RPL mais cette fonction invalide le parent préféré sans le supprimer. Si bien que le DODAG se reconstruit en utilisant l'adresse du parent préféré initial.
Finalement, afin d'empêcher ce comportement, nous forçons les nœuds à mettre à jour leur parent préféré en supprimant tous les parents choisis par un nœud après le changement de pseudonymes. Cette solution n'apporte pas de surcoût de trames, ni n'introduit de modifications indésirables des tables. De cette façon, le comportement naturel de RPL est préservé, les nœuds peuvent choisir leur parent préféré pendant la mise à jour du DODAG. Seule une étape supplémentaire est ajoutée à l'implémentation de [115] sans introduire de modification du comportement prévu. Nous avons donc pu valider le comportement décrit dans [115]. Une autre solution pourrait être d'enrichir la simulation en autorisant la modulation des métriques radios. Pour cela, il serait intéressant d'utiliser le modèle de couche radio fourni par WSNET à la place de notre couche posix radio.

Le deuxième problème dû à l'implémentation concerne également RPL et son fonctionnement par défaut dans Contiki et va impacter les performances de MT6D. Dans un déploiement réel, lorsque un nœud apparait, disparait ou est en mouvement, les routes sont mises à jour de manières différentes.
Quand un nouveau nœud rejoint le réseau, une nouvelle entrée le concernant est ajoutée dans la table de routage de tous les nœuds concernés. Ainsi, dans la Figure 7.2, "B" en rejoignant le réseau a ajouté une route le concernant dans la table de routage de "A" et du 6BR. En revanche, si un nœud venait à se déplacer ou disparaitre, les nœuds qui avaient un chemin par lui reconstruisent un nouveau chemin à travers un autre parent et les routes seraient mises à jour. Par exemple, si "B" disparait, "C" peut choisir de passer par "A" (si celui-ci est à sa portée). Dans ce cas, le 6BR mettra à jour la route pour atteindre "C" dans sa table de routage.
Dans MT6D, le changement de pseudonymes apparait pour les voisins comme l'ajout d'un nouveau nœud (car nouvelle adresse). La table de routage est donc mise à jour avec chaque nouveau pseudonyme. De plus, conformément à l'implémentation décrite dans l'article [115], la durée de vie du DAO est mise à une valeur infinie. Un élément de la table de routage n'est donc supprimé que si la taille maximale est atteinte. Néanmoins, les auteurs indiquent qu'il est possible de changer cette valeur pour un temps fini. Ainsi, si aucun DAO n'est reçu pendant un certain laps de temps, la route est supprimée de la table de routage. De même, RPL spécifie qu'un nœud peut être supprimé de la table des voisins si aucun DIO n'est reçu de sa part pendant une certaine période. Néanmoins, dans la version par défaut de Contiki, cette caractéristique n'est pas implémentée. Les parents tout comme les routes ne sont donc jamais supprimés. Lors du déploiement de MT6D, à chaque changement de pseudonymes, les tables utiles au routage vont donc grossir. Elles seront totalement reconstruites à chaque changement de pseudonymes même si le réseau n'évolue pas. De nombreuses entrées ne seront plus valables. Lorsque le maximum d'entrées est atteint, les voisins les plus anciens sont supprimés afin de libérer de la place pour les nouveaux. Le même procédé est utilisé sur la table de routage. Le but n'est pas de modifier Contiki ou le fonctionnement de RPL. Le déploiement de MT6D doit pouvoir se faire de manière à être comparé au déploiement d'Ephemeral. Il a donc été choisi de ne pas déployer de solutions pour contrer ce manque dans Contiki. Le comportement des tables de routage pour Ephemeral sera à regarder vis-à-vis de ce problème apporté par MT6D.

Annexe D

Implémentation d'Ephemeral dans Contiki

D.1 Exemple d'utilisation d'Ephemeral

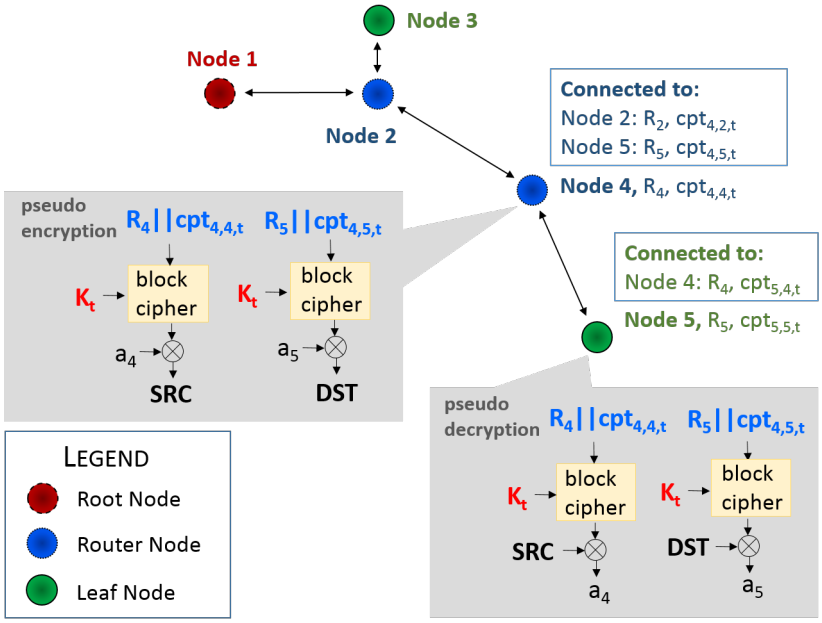


FIGURE D.1 – Exemple d'utilisation d'Ephemeral.

Voici un exemple d'un réseau implémentant Ephemeral.

La topologie du WSN est donnée dans la Figure D.1. Lors de l'initialisation, chaque nœud du réseau reçoit la clé secrète K_t utilisée pour générer les pseudonymes. Il reçoit ou génère également son R. Puis, chaque nœud stocke l'adresse MAC a_j de ses q voisins ainsi que le R_j correspondant. Pour le nœud 4, cette table comporte 2 entrées alors que pour le nœud 5 seule 1 entrée est présente. La taille maximale de la table est identique à celle de la table des voisins. Cette table enregistre également la valeur du compteur courant $\text{cpt}_{i,j,t}$ pour chaque communication *hop-by-hop*. Quand le nœud 4 envoie une trame Ephemeral au nœud 5, il calcule les pseudonymes à utiliser pour les adresses MAC source et destination.

Pour cela, le nœud 4 utilise son matériel de sécurité ($\text{cpt}_{4,4,t}$ et R_4) mais également le matériel de sécurité associé au nœud 5 ($\text{cpt}_{4,5,t}$ et R_5). Lors de la réception, le nœud 5 déchiffre le pseudonyme destination en utilisant la valeur R_5 stockée dans sa table utilisée pour Ephemeral et le compteur $\text{cpt}_{4,5,t}$ inclut dans

l'en-tête de la trame. Il regarde alors si la valeur obtenue correspond à son adresse MAC réelle et si la valeur des compteurs ne diffère pas trop de celles qu'il stocke dans sa table. Si tout est bon, il peut alors déchiffrer la trame.

Nous avons donc déployé Ephemeral dans Contiki OS. Contrairement à MT6D, Ephemeral ne remplace pas définitivement l'adresse MAC du nœud par un pseudonyme. Il garde en mémoire la valeur initiale (réelle) de son adresse. La phase de *setup* reproduite par MT6D à chaque nouveau pseudonyme est donc inutile. Le nœud sera toujours identifié pour le routage par ses adresses réelles. Cet avantage pour la construction et le maintien des tables de routage impacte le bon déroulement de plusieurs protocoles qui utilisent les informations d'adresses pour créer ou parser une trame et donc plusieurs couches du modèle OSI. De plus, de nouvelles structures ont dues être mises en place.

Des choix d'implémentation ont donc dû être faits.

D.2 Implémentation dans Contiki 3.0

Dans cette partie nous allons détailler l'implémentation dans Contiki.

Chaque nœud a besoin d'une nouvelle structure pour stocker le matériel de génération des pseudonymes. Par la suite, cette table qui stocke le matériel utilisé dans Ephemeral pour la génération et la vérification des pseudonymes sera appelée table de "privacy". Contiki utilise un format spécifique pour ses tables de routage. La table de "privacy" va s'inspirer de ce format pour stocker les informations nécessaires. Dans ce format, la table est indexée par les adresses MAC. A chaque adresse est associée une structure incluant l'IV et le compteur courant. Cela facilite la recherche lors des générations de pseudonymes. Il est ainsi possible de réutiliser les fonctions définies dans Contiki pour extraire un attribut ou ajouter un nouvel élément. La table est également limitée à 30 entrées.

Le matériel de génération des pseudonymes propre au nœud est stocké dans une structure indépendante. La partie R de l'IV$_t$ du nœud est générée grâce à l'utilisation d'un générateur de nombre pseudo aléatoire : l'algorithme "XORSHIFT128". Cet algorithme est appelé avant l'association du nœud au réseau. Il est également appelé lorsqu'il est nécessaire de mettre à jour la valeur de l'IV$_t$. La fréquence de régénération de R dépend du choix de périodicité du changement de pseudonyme.

Pour rappel, Ephemeral autorise deux types de planification :

- Sur évènement : le compteur est incrémenté à chaque utilisation. Ainsi, chaque pseudonyme ne sera utilisé qu'une seule fois même lors du routage d'une trame. Néanmoins, cette planification nécessite un coût de renouvellement de la clé K_t et des différents IV important et peut permettre à un attaquant d'utiliser le compteur comme identifiant de l'activité d'un nœud. En effet, un nœud routeur proche du puits doit émettre plus de trames. Son compteur est donc incrémenté plus souvent qu'un nœud feuille.
- Périodiquement : les compteurs sont mis à jour à intervalles de temps réguliers. Les nœuds peuvent être désynchronisés i.e. les compteurs peuvent ne pas avoir tous la même valeur, les communications avec Ephemeral pourront avoir lieu.

Dans notre implémentation, nous avons choisi de mettre en place la planification périodique. L'IV$_t$ n'est remis à jour que lorsque le compteur atteint sa valeur maximale.

Pour rejoindre le réseau, le nœud va ensuite lancer le protocole d'association défini dans le standard RPL. L'une des différences fondamentale de l'utilisation d'Ephemeral est que le nœud ne va utiliser les adresses MAC réelles que pour l'émission du premier DIS. Toutes les trames échangées par la suite utiliseront des pseudonymes. Durant cette association, les tables de routage seront construites avec les adresses réelles des nœuds du réseau même si les trames échangées utilisent des pseudonymes.

Les nœuds gardent en mémoire leur adresse MAC liée au matériel et ne génère les pseudonymes que pour construire l'en-tête MAC. Ce comportement facilite le maintien des tables de routage ainsi que la gestion des adresses IPv6 lors du changement de pseudonymes. Néanmoins, cela entraine des modifications pour la couche radio. La différence entre l'adresse MAC destination et l'adresse MAC initiale mémorisée du nœud récepteur entraine un dysfonctionnement du filtrage d'adresse.

Considérons le nœud i dont l'adresse MAC mémorisée initiale est a_i. Avec Ephemeral, la radio utilise a_i pour le filtrage d'adresse. Lorsqu'un nœud j souhaite communiquer avec i, il va établir le pseudonyme destination p_i qu'il va indiquer dans l'en-tête MAC. Si le comportement n'est pas modifié, la radio du nœud i reçoit alors une trame indiquant comme adresse destinataire p_i. Comme $p_i \neq a_i$, la radio va filtrer la trame. Une première idée était de calquer le comportement de MT6D et donc d'indiquer à la radio que l'adresse MAC pouvait prendre une autre valeur, celle du pseudonyme. Avec cette solution, les nœuds sont alors obligés d'être synchronisés et l'on perd l'avantage qu'offre Ephemeral. Le mode promiscuité a donc été mis en place afin de supprimer le filtrage d'adresse de la couche radio et de laisser ce travail à la couche MAC.

Il faudra tout de même faire attention car cette modification concerne la radio. Or, celle-ci est dépendante de la plateforme. En effet, cette modification n'est pas nécessaire pour la simulation car la couche radio n'effectue pas de filtrage d'adresse. Néanmoins, celle-ci sera obligatoire pour une implémentation en réel et devra prendre en compte les spécifications de la puce radio. Toutes les autres fonctions utilisées par Ephemeral peuvent être portées sur n'importe quelle plateforme.

Pour vérifier le pseudonyme d'un émetteur, le nœud a besoin de connaitre l'IV qui a permis de le générer. Il doit également distribuer son R qu'il a généré. Ephemeral utilise RPL pour diffuser les éléments publiques du protocole. En effet, en plus de l'association, RPL défini un envoi périodique de trames ICMPv6 pour maintenir le routage. L'association et le maintien périodique vont permettre la diffusion et la mise à jour des R_i du réseau. Dans les trames DIO et DAO, RPL fournis une implémentation de champs optionnels inutilisés. Le protocole RPL est enrichi d'une nouvelle option appelée "RPL_OPTION_IV" qui permet d'inclure R dans ces champs. Chaque nœud diffuse à ses voisins sa valeur courante de R à chaque fois que RPL est exécuté.

De cette façon, Ephemeral n'introduit pas de trames de contrôle supplémentaires pour diffuser les éléments publics nécessaires à son enrichissement. Néanmoins, la taille des trames DIO/DAO est impactée. Afin de permettre le traitement et la récupération de l'IV_t, un en-tête de 8 bits est ajouté pour indiquer la nouvelle option du message ICMP. Un autre en-tête de 8 bits permet d'indiquer la taille du payload de l'option. La taille de R va dépendre de la taille du cpt_t utilisé. En effet, si cpt_t est de taille 8 bits, R introduira un surcoût de 17 octets. Si cpt_t est de taille 16 bits, le surcoût sera ramené à 16 octets. Le même mécanisme est utilisé pour mettre à jour la valeur de R.

Du fait qu'Ephemeral se base sur RPL pour le partage des R_i, nous devons faire attention à l'intervalle maximum entre deux trames DIO. En effet, si le compteur atteint sa valeur maximale, R doit être mis à jour. Or ce nouveau R doit être inclus dans le prochain DIO ou DAO. Si l'envoi du DIO n'intervient pas avant l'incrémentation du compteur, le nœud devra alors utiliser son R actuel avec une valeur de compteur déjà utilisée. Afin d'éviter l'utilisation de cette information par un attaquant, une marge de sécurité "safety_margin" doit être définie vis-à-vis de l'intervalle maximal entre deux DIO défini dans Contiki et la périodicité de l'incrémentation.
Imaginons que la taille du compteur est $\ell_2 = 8$ bits. 255 incréments sont alors possibles. Prenons une périodicité de 60s. Contiki défini un intervalle maximal d'environ 18 minutes entre chaque protocole périodique de maintien des tables. Il est donc nécessaire de commencer le processus de régénération de R au moins quand cpt_t atteint la valeur $255 - 18 = 237 = safety_margin$.
Les réseaux IEEE 802.15.4 sont des réseaux où des paquets peuvent être perdus. Un voisin peut donc perdre un DIO et ne pas avoir la mise à jour du R. Dans ce cas, l'application sera interrompue jusqu'au nouvel envoi d'un DIO contenant le R. Contiki défini une valeur maximale mais également une valeur minimale pour l'émission des DIO. Ces valeurs peuvent être adaptées au cas d'usage afin de limiter la perte de trames d'application.

La Table D.1 détaille la probabilité d'une collision calculée avec la formule de Zenner dans le cas où la mise à jour de R a lieu pour chaque DIO émis. Le calcul est réalisé pour les valeurs minimale et maximale définies dans Contiki.

DIO interval	nb of reset	p_{max}
4s	78750000	$\approx 2^{-71}$
18 min	308 823	$\approx 2^{-87}$

Table D.1 – Probabilité de collision.

Dans le cas d'un réseau fonctionnant 10 ans avec 30 nœuds, même lorsque l'envoi de DIO est programmé pour la périodicité minimale, la probabilité de collision d'IV_t est assez faible pour ne pas nécessiter la régénération de la clé K_t.

Grâce à RPL, le matériel nécessaire à l'exécution d'Ephemeral peut donc être distribué sans ajout de trames.

La table de "privacy" complétée avec les valeurs des R des différents voisins, un nœud peut alors vérifier les pseudonymes sources et destinations des messages entrants.

Afin de fonctionner correctement, Ephemeral a besoin d'ajouter des champs dans l'en-tête MAC IEEE 802.15.4. Cet ajout doit être fait en concordance avec le standard. L'amendement IEEE 802.15.4e de 2012 introduit les champs "Information Element" comme expliqué dans la partie 3. Ces champs supplémentaires permettent l'extension de l'en-tête MAC. Pour le moment la version de Contiki ne permet pas d'utiliser ces champs. Nous avons donc dû modifier le format et le traitement de l'en-tête MAC pour prendre en compte les "Information Element". Les modifications ont été faites de façon à permettre une rapide transposition lorsque la version 2012 sera déployée dans Contiki. Pour cela, le standard a été étudié et les spécifications ont été suivies.

Ephemeral est donc implémenté de façon à permettre à un nœud en mobilité de récupérer le R d'un nouveau voisin, contenu dans un DIO (ou DAO), qui utilise un pseudonyme. Ephemeral permet également de construire les tables de routage avec l'adresse réelle de ce nouveau voisin et non avec le pseudonyme utilisé.

Ephemeral, par l'utilisation des DIO/DAO pour propager les IV permet à un nouvel entrant de récupérer les valeurs des R, de retrouver l'adresse réelle et de construire les tables via cette adresse.

Pour cela, le traitement des trames RPL a dû être modifié. En effet, dans Contiki, un nœud construit sa table de routage avant de traiter le contenu de la trame RPL. Dans ce cas, lorsqu'un nouveau nœud recevait une trame DIO (resp. DAO), il ne possédait le matériel nécessaire pour déchiffrer le pseudonyme que lorsqu'il avait fini de parser le contenu. La table étant enregistrée avant, l'adresse était donc un pseudonyme.

Ephemeral déplace cette fonction. Il autorise le nœud à ne mettre à jour sa table des voisins qu'après avoir parsé le message. Ainsi, le nœud va retrouver l'adresse réelle grâce à la valeur de R contenue dans la trame RPL puis mettre à jour sa table avec cette adresse et non avec le pseudonyme.

Les nœuds ont maintenant tout le nécessaire pour communiquer en utilisant des pseudonymes. L'utilisation et la distribution des R mais également l'incrémentation des cpt_t ou encore la nature du réseau entrainent des cas critiques qu'il faut résoudre dans l'implémentation.

D.3 Cas critiques

D.3.1 Impact sur la QoS

Quand il y a perte d'un paquet contenant la nouvelle valeur de R, suivant le type de communication, ainsi que la position du nœud dans le DODAG, l'impact sur la qualité de service ne sera pas identique.

Si le nœud B (Figure 7.2) ne reçoit pas le DAO contenant le R du nœud C, cela n'impacte pas les communications *upward*. En effet, C connait le R de B et peut donc construire son pseudonyme. Il envoie sa trame à B qui la transmettra à A sans se soucier de C. En revanche, pour les communications *downward*, les messages seront filtrés car le pseudonyme indiqué en adresse destination dans la trame comme adresse de C aura été généré à partir d'un IV non valide. C n'arrivera pas à partir de son IV et de la valeur du compteur fournie à retrouver son adresse MAC initiale. C pensera alors que le message ne lui est pas adressé. Le bon IV n'ayant pu être récupéré, les communications sont alors perturbées dans le cas d'une perte d'un DIO et/ou DAO. Il faudra attendre le prochain RPL périodique pour mettre à jour les valeurs des R avec les nouvelles valeurs transmises. On a donc, en cas de perte de paquet, une perte de trames maximale de :

$$max = \frac{MAX_RPL}{p}$$

avec MAX_RPL la périodicité maximale définie dans Contiki et p la périodicité d'émission de trame d'application.

D.3.2 Mise à jour de R et problèmes de compteurs

Un autre cas critique de l'IV concerne également les compteurs. Prenons un compteur sur 8 bits. La valeur maximale de ce compteur est de 255. Considérons ren comme marge de sûreté et considérons $ren = 3$. Si le compteur d'un nœud ou le compteur stocké d'un de ses voisins atteint $255 - ren = 252$, un mécanisme est lancé pour forcer la régénération d'un nouveau R. Ainsi, si le compteur du nœud atteint 252, le nœud force l'émission d'un DIO (ou DAO si nœud feuille) contenant une version nouvelle de R. En revanche, si le nœud identifie l'un des compteurs stockés pour ses voisins comme valant 252, le nœud averti ce voisin par l'émission d'une trame UDP indiquant le besoin de générer un nouveau R.

Dans le cas où l'on ne reçoit pas de nouveau R à temps car la trame a été perdue ou le nœud n'a pas régénéré comme demandé, les compteurs stockés sont incrémentés de façon à ne jamais boucler. En effet, en langage C, si, par exemple, le compteur est sur 8 bits, la valeur maximale est donc 255. Lors du prochain incrément, cette valeur retombe à 0. Pour les raisons de sécurité expliquées précédemment et parce qu'un compteur à 0 indique l'utilisation de l'adresse MAC réelle, quand la valeur maximale est atteinte, le prochain incrément fait que le compteur reste à cette valeur maximale. Ainsi, même si un nœud utilise plus longtemps un pseudonyme, il ne casse pas la protection de la vie privée en utilisant un compteur à 0. De plus, cette vérification est effectuée avant la régénération forcée. Le mécanisme sera donc appelé pour régler le problème.

Nous avons vu que les compteurs envoyés en clair pouvait représenter un identifiant unique pour un attaquant. Nous avons donc évoqué la synchronisation des nœuds. Pour ce faire, un nœud compare le compteur source reçu dans la trame aux compteurs stockés. Si cette différence est supérieur à la valeur de δ définie, le compteur stocké est alors mis à jour avec la valeur reçue. La nouvelle valeur du compteur n'est jamais inférieure à la valeur courante du compteur de façon à éviter la réutilisation d'un même IV.

De plus, la mise en place de ce nouveau protocole ne doit pas nuire aux protocoles déjà déployés. Le mécanisme de synchronisation modifie les valeurs des compteurs pouvant nuire au protocole de renouvellement des R.

Lors de la synchronisation, l'implémentation doit donc prendre en compte la marge de sécurité "safety_margin" définie précédemment pour le bon déroulement du protocole de régénération de l'IV. La condition suivante est donc testée :

$$if((cpt_recu > safety_margin)\&\&(cpt_stocke < safety_margin))$$
$$cpt_stocke = safety_margin;$$

avec cpt_recu : la valeur du compteur source de la trame, "safety_margin" la marge laissée pour la régénération (calculée à 237 pour un compteur 8 bits) et cpt_stocke la valeur du compteur stocké.

Dans ce cas, on met à jour les compteurs avec la limite basse permettant l'émission au prochain tour d'un nouveau R. On autorise donc la synchronisation sans perturber la régénération. En revanche, si $(cpt_stocke > safety_margin)$ alors on ne change pas la valeur des compteurs stockés pour ne pas perturber le mécanisme de régénération. Le changement de R pourra donc intervenir. Néanmoins, il y aura de la désynchronisation. Dans notre implémentation nous avons choisi de privilégier la bonne diffusion des IV.

Ce mécanisme est également utilisé pour synchroniser le compteur destination reçu avec la valeur du compteur du nœud destinataire de la trame.

D.3.3 Cohabitation avec les mécanismes déjà déployés

La cohabitation d'Ephemeral avec les protocoles de sécurité MAC a posée problème. En effet, lorsqu'un nœud met en place le chiffrement MAC, il utilise son adresse source pour fabriquer l'IV. Cette méthodologie est utilisée afin de réduire les données à stocker dans la mémoire du nœud pour le déploiement de la sécurité. Un nœud doit alors simplement connaitre la clé de chiffrement. Le reste des informations est transmis dans

les différents champs de l'en-tête MAC. Or, dans le fonctionnement classique, c'est l'adresse réelle qui est utilisée pour fabriquer une partie de l'IV comme expliqué dans la partie 3. Un nœud n'appartenant pas encore au réseau et souhaitant le rejoindre ne connait pas cette adresse réelle car il reçoit une trame avec des pseudonymes. De plus, il ne peut la retrouver car il ne possède pas le matériel. Il lui est donc impossible de déchiffrer la trame pour pouvoir accéder au matériel utilisé par Ephemeral qui pourrait lui permettre de retrouver la bonne adresse. Il est donc bloqué. Ce problème se pose également lorsqu'un nœud est mobile dans le réseau et qu'il change de parent. Avec Ephemeral, le chiffrement d'une trame à émettre n'est plus réalisé avec un IV basé sur l'adresse réelle du nœud mais basé sur le pseudonyme transmis dans la trame. Ainsi, même si un nouveau nœud ne possède pas le matériel pour obtenir l'adresse réelle, il va pouvoir déchiffrer les trames d'association et en extraire la valeur de R. Il peut ensuite retrouver l'adresse réelle du nœud émetteur et construire ses tables de routage. De cette façon, on ne modifie pas la construction de l'IV défini dans le standard IEEE 802.15.4 et l'on permet aux nouveaux nœuds ou aux nœuds mobiles de rejoindre le réseau.

Les mécanismes de compression d'adresses ont également posés problème avec Ephemeral. En effet, lors de l'association d'un nouveau nœud, celui-ci ne connait pas encore les IV utilisés par ses voisins. Il lui faut donc pouvoir accéder au matériel contenu dans les champs ICMPv6 des trames RPL échangées pendant le protocole d'association. Dès lors où le nouveau nœud récupère les IV de ses voisins, il peut créer ses tables de routage avec les adresses MAC réelles. Or, lorsque la trame est émise sans saut, par exemple dans le cas d'un broadcast d'une trame DIO, le mécanisme de compression de 6LoWPAN autorise à compresser l'adresse IPv6. Dans ce cas, la trame ne comprend pas de champ d'adresse IPv6 destination. Le nœud destinataire déduit que l'adresse est formée grâce à SLAAC et à l'adresse MAC source contenue dans la trame. Or, si le nouveau nœud n'a pas pu retrouver l'adresse MAC réelle car il ne possède pas encore le matériel, l'adresse IPv6 déduite ne sera pas la bonne mais correspondra à celle obtenue via SLAAC et le pseudonyme. Il ne pourra donc pas créer de tables de routage contenant les adresses réelles de ses voisins. De plus, cette adresse IPv6 est utilisée pour calculer le *checksum* de la couche ICMP. En effet, ce *checksum* calculé à la couche transport permet d'éviter les erreurs de transmission. Il permet notamment de détecter une erreur sur l'adresse destination qui entraine un mauvais routage de la trame. Le destinataire du nœud doit alors vérifier les informations de l'en-tête IPv6. C'est pourquoi le *checksum* de la couche transport inclut les informations de la couche IP. Lors du déploiement d'Ephemeral, de par l'ordre de traitement des couches pour une trame émise, le nœud source calcule ce *checksum* avant la génération de son pseudonyme (couche transport traitée avant couche MAC). Le *checksum* est alors calculé avec l'adresse IPv6 formée grâce à l'adresse MAC réelle. Lors du protocole d'association, le nouvel arrivant ne connait pas l'adresse MAC réelle ni les IV. Il traite la trame en partant de la couche MAC et en remontant à la couche Application. Au moment du traitement de la couche IP, le nouveau nœud considère l'adresse MAC transmise dans la trame comme adresse réelle car il ne possède pas d'IV pour ses voisins. Du fait du mécanisme de compression, le nœud déduit donc l'adresse IPv6 source comme liée à ce pseudonyme. Le *checksum* calculé par le nœud destinataire est donc faux car ne correspond pas à celui calculé et transmis par le nœud source de la trame. Le nouvel arrivant considère la trame comme mauvaise et le traitement est avorté. Ce dernier ne pourra alors jamais atteindre les champs ICMPv6 et récupérer les informations nécessaires à Ephemeral. Afin de pallier à ce problème, lors de la réception d'une trame, le contrôle du *checksum* est désactivé temporairement. Il ne faut néanmoins pas le désactiver pour toutes les trames reçues. Il est nécessaire de laisser le mécanisme de *checksum* identifier une trame transmise avec des erreurs de bits permettant l'avortement du traitement de celle-ci. La désactivation ne doit intervenir que lorsqu'il est probable que le nœud n'a pu retrouver l'adresse réelle associée au pseudonyme car ne possédant pas les IV associées à l'émetteur. Cette désactivation temporaire permet au nœud de récupérer les IV pour retrouver ensuite l'adresse réelle de l'émetteur de la trame. Néanmoins, il ne faut pas totalement supprimer ce calcul de *checksum*. Il est donc recalculé après la récupération du matériel et le calcul de l'adresse réelle. Ce changement de fonctionnement doit être analysé vis-à-vis de la sécurité du réseau. Face à un attaquant externe, cela n'a pas d'impact car la sécurité MAC l'empêche de forger une trame valide. Le traitement de celle-ci sera donc avorté bien avant d'atteindre la couche ICMP. Lorsque le réseau fait face à un attaquant interne, celui-ci peut premièrement utiliser ce mécanisme afin de mener une attaque *poisoning*. Il va alors forger des trames de contrôle dont le pseudonyme MAC source n'est pas liée à une adresse réelle (valeur aléatoire) et où le reste de la trame est correct (*checksum* ICMP compris). Le nœud récepteur va alors traiter cette trame comme un nouvel arrivant désactivant temporairement le *checksum*. Il va ensuite ajouter une fausse adresse dans ses tables de routage. Cela va entrainer un surcoût de mémoire. Néanmoins, cette attaque est réalisable même lorsque le *checksum* fonctionne de manière classique. L'attaquant connaissant le matériel

de protection de la vie privée et de sécurité peut aisément forger des trames en falsifiant des identités sans compter sur ce mécanisme de désactivation de *checksum*. Il peut alors mener des attaques similaires aux attaques *overflow* et *poisoning* réalisées sur les tables de routage. Dans ces attaques, de fausses routes sont ajoutées aux tables par un attaquant interne comme l'expliquent Abdelaziz, Nafaa et al. dans [123]. La deuxième attaque interne peut permettre une exhaustion de la batterie. Il va forger la même trame que précédemment mais où le *checksum* ICMP est faux. Le nœud ne le calculera qu'après traitement de la couche RPL. Le nœud aura donc effectué plus d'opération avant de rejeter la trame. Là encore, un attaquant interne peut mener une attaque par exhaustion de batterie même en l'absence de la solution pour le *checksum*. Les attaques par exhaustion de batteries sont difficiles à contrer.