Détection d'intrusions réaliste dans les maisons connectées à l'aide d'indicateurs physiques volatiles

Olivier Lourme

UNIVERSITÉ DE LILLE

École doctorale **MADIS-631**

Unité de recherche **CRIStAL**

Thèse présentée par **Olivier LOURME**

Soutenue le **21 décembre 2023**

En vue de l'obtention du grade de docteur de l'Université de Lille

Discipline **Informatique**

Spécialité **Informatique et applications**

Détection d'intrusions réaliste dans les maisons connectées à l'aide d'indicateurs physiques volatiles

Thèse dirigée par Michaël HAUSPIE

Composition du jury

Rapporteurs	Hervé DEBAR	Professeur à Telecom SudParis
	Benoit PARREIN	MCF HDR à Nantes Université
Examinateurs	Nathalie MITTON	Directrice de recherche à l'INRIA, Présidente du jury
	Antoine GALLAIS	Professeur à l'Université Polytechnique Hauts-de-France
	Gilles GRIMAUD	Professeur à l'Université de Lille
Directeur de thèse	Michaël HAUSPIE	MCF HDR à l'Université de Lille

Remerciements

En septembre 2019, j'ai pris la décision de démarrer une thèse portant sur l'Internet des objets et sa sécurité. Jusqu'alors enseignant de Génie Électrique et d'Informatique en IUT, ma curiosité m'avait toujours poussé à développer des projets associant informatique et électronique, que je réinvestissais dans mes enseignements. L'envie d'acquérir de nouvelles compétences, le souhait de voir comment la connaissance se constitue et la motivation de relever un défi personnel m'ont amené à quarante-sept ans à redevenir étudiant pour préparer un doctorat. Ce ne fut pas un long fleuve tranquille et je tiens à remercier ici les personnes qui ont rendu cette aventure possible.

Je remercie Hervé Debar et Benoît Parrein pour l'honneur qu'ils m'ont fait en acceptant de rapporter cette thèse. Leurs retours nourris m'ont permis de prendre de la hauteur sur mon propre travail et je garde précieusement leurs remarques pour continuer de faire évoluer mon sujet. Je remercie également les examinateurs Nathalie Mitton, Antoine Gallais et Gilles Grimaud pour l'attention qu'ils ont portée à mon travail, ainsi qu'en témoigne leur intérêt pour la soutenance.

Je remercie à nouveau Gilles Grimaud, chef de l'équipe 2XS du laboratoire CRIStAL, d'abord pour m'avoir accepté dans son équipe il y a quatre ans puis pour m'avoir expliqué à de nombreuses reprises ce qu'était une thèse et une démarche scientifique. Je lui sais gré également de s'être penché régulièrement sur mes travaux de façon informelle et d'avoir à ces occasions apporté un supplément d'âme à mes réflexions et analyses.

J'adresse un grand merci à Michaël Hauspie, mon directeur de thèse, qui a toujours fait preuve d'une grande disponibilité et d'une grande écoute. Ses conseils d'orientation lors du chemin incertain du doctorat m'ont été précieux. En outre, ses expertises sur les outils du chercheur et sur ceux plus spécifiques des thématiques que j'ai abordées m'ont permis d'appréhender le fonctionnement de la recherche et aussi d'envisager des expériences auxquelles je ne pensais même pas. Il m'a également fait confiance en me permettant d'encadrer des travaux de Master 2 et en me confiant la responsabilité de la session doctorants de RESSI 2023.

Merci également aux maîtres de conférences Thomas, Alexandre et Pierre pour leur gentillesse et leurs conseils pleins d'expérience.

Merci à mes co-doctorants qui m'ont accompagné et permis de me sentir toujours jeune : Étienne, formateur ès radios logicielles au début de ma thèse, Florian, mainteneur de liens, Nicolas, alter-ego utcéen et Clément, blagueur émérite, synchrone avec moi jusqu'à la semaine de soutenance de thèse ! Merci aussi à Damien, ingénieur de recherche, avec qui nos échanges tôt le matin ont installé un point de repère agréable.

Je me dois aussi de remercier Denis Pomorski et David Le Toriellec respectivement Directeur de l'IUT de Lille et Chef du Département GEII où j'ai continué de réaliser mon enseignement pendant la thèse. Ils ont toujours émis des avis très favorables à mes demandes d'aménagement de service auprès de l'Université de Lille, qui a ainsi pu me financer pendant quatre ans 192 heures de décharge. Je sais gré à cette dernière de m'avoir procuré de telles conditions, me donnant le rare privilège de fréquenter la recherche scientifique. Je remercie tous mes collègues de l'IUT pour leurs encouragements et leurs questions régulières sur l'avancement de mes travaux. Je remercie notamment Cyrille qui s'est arrangé pour me façonner un emploi du temps en phase avec le calendrier de la thèse. Je remercie également Jean-Marie Place, dévoué collègue qui m'a introduit auprès de 2XS.

Pendant plus de quatre ans, cette thèse a quelque peu bouleversé ma vie et je n'aurai pas pu la mener à bien sans le soutien au long cours de mes amis et de ma famille.

Merci aux amis de St-Pol et à ceux de l'UTC pour leur intérêt dans mes recherches. Merci aux amis coureurs du Héron pour la soupape sportive indispensable du samedi matin.

Je remercie ici mes parents pour m'avoir transmis le goût du travail et pour leur soutien sans faille. Je remercie mon frère, ma sœur et leurs familles pour leurs encouragements. Merci aussi à ma belle-famille. Tous, vous m'avez permis d'avancer.

Enfin, je dis un grand merci à mes deux filles Anna et Jane qui dès le CM1 ont vu un papa beaucoup moins présent mais qui en ont profité pour devenir autonomes et responsables. Je les adore, elles sont drôles et uniques. Je leur souhaite d'être heureuses dans leurs vies. Le dernier merci est pour mon épouse, Maria, complice stimulante et indéfectible depuis dix-huit ans. Sans elle et son amour, ce voyage initiatique n'aurait pas été possible. *b7ébb-ik ktîr ya 7abîbt-é !*

À ma famille

Parvenu à l'issue de mon doctorat en informatique, et ayant ainsi pratiqué, dans ma quête du savoir, l'exercice d'une recherche scientifique exigeante, en cultivant la rigueur intellectuelle, la réflexivité éthique et dans le respect des principes de l'intégrité scientifique, je m'engage, pour ce qui dépendra de moi, dans la suite de ma carrière professionnelle quel qu'en soit le secteur ou le domaine d'activité, à maintenir une conduite intègre dans mon rapport au savoir, mes méthodes et mes résultats.

SERMENT DOCTORAL D'INTÉGRITÉ SCIENTIFIQUE

Nous sommes des nains assis sur des épaules de géants. Si nous voyons plus de choses et plus lointaines qu'eux, ce n'est pas à cause de la perspicacité de notre vue, ni de notre grandeur, c'est parce que nous sommes élevés par eux.

Bernard de Chartres, maître du XII^{e} siècle

Résumé en français

Détection d'intrusions réaliste dans les maisons connectées à l'aide d'indicateurs physiques volatiles

Au sein de l'Internet des Objets, le secteur de la maison connectée est en plein essor. Pour quelques dizaines d'euros, chacun peut s'équiper de solutions domotiques intelligentes commandables à distance. Ces écosystèmes sont cela dit vulnérables à des attaques variées en raison A) d'une conception essentiellement guidée par le coût, générant des objets contraints sans implémentation de sécurité viable possible, B) de l'utilisation par ces objets de protocoles de communication sans-fil hétérogènes, dispersant les efforts de sécurisation, et C) de la gestion de ces objets par des consommateurs non-experts, adeptes du « setup and forget ».

Contrairement à l'informatique traditionnelle où les solutions de protection sont répandues, nous faisons le constat de l'absence de proposition commerciale équivalente dans la maison connectée. Dans cette thèse, nous nous interrogeons sur les conditions de l'adoption à grande échelle de solutions de sécurité de type Systèmes de Détection d'Intrusion (IDS), visant à protéger les objets contraints déjà déployés. Ainsi, une première contribution recense les caractéristiques des maisons connectées pour les croiser avec des taxonomies d'IDS, afin de proposer les critères qualitatifs d'une solution de sécurité domestique réaliste.

Par la suite, afin de faciliter la conception d'IDS, une deuxième contribution met à disposition de la communauté scientifique un jeu de données Zigbee, participant à la fourniture d'outils couvrant les principaux protocoles de la maison connectée. Toutes les trames échangées par 10 objets pendant 10 jours ont été capturées par 4 sondes distribuées dans un domicile-test. Des attaques ont été introduites afin d'établir et comparer différentes stratégies de détection. Outre une redondance des données de couche MAC, le jeu de données tire son originalité de l'extraction par chaque sonde du RSSI (*Received Signal Strength Indicator*) de chaque trame. Cette grandeur de couche physique, accessible à peu de frais dans les technologies sans-fil, permet de participer à l'identification de nœuds fixes. Par la suite, on peut imaginer d'identifier robustement chaque objet par une empreinte de couche physique faite d'un tuple de RSSI, complexe à imiter par un attaquant.

Enfin, dans une troisième contribution, nous exploitons le jeu de données pour proposer un IDS détectant les attaques d'usurpation d'identité, favorisées par le fait que des piles de protocoles n'intègrent que peu ou pas d'authentification sur leur couche MAC. Pour les détecter, la cohérence de l'identifiant de couche MAC et de l'empreinte précédente à base de RSSI peut être considérée mais ce n'est plus possible quand les environnements sont sans cesse redessinés par les habitants qui y évoluent, les RSSI devenant volatiles. En fournissant des séries temporelles de RSSI en entrée d'un algorithme d'apprentissage non-supervisé, nous établissons pour chaque couple (objet, sonde) un modèle des séquences RSSI normales. Les déviations par rapport au modèle permettent de détecter une attaque. Les métriques de détection obtenues, très intéressantes en regard de la faible complexité de l'architecture initiale envisagée, ainsi que les évaluations de l'autonomie et du coût de la solution laissent entrevoir une diffusion de tels systèmes dans les maisons connectées.

Mots-clés : Sécurité de l'Internet des Objets, Systèmes de détection d'intrusion, Jeux de données, Attaques d'usurpation d'identité, Apprentissage profond, Zigbee.

Résumé en anglais (abstract)

Realistic intrusion detection in smart homes using volatile physical features

Within the Internet of Things, the smart home sector is booming. For a few tens of euros, everyone can be equipped with smart-home automation solutions that can be controlled remotely. However, these ecosystems are vulnerable to various attacks due to A) an essentially cost-driven design, generating constrained devices with too few resources for viable security implementations, B) the use by these devices of multiple wireless communication protocols, dispersing security efforts, and C) the management of these devices by non-expert consumers, following a “setup and forget” policy.

Unlike traditional IT where protection solutions are widespread, we note the absence of an equivalent commercial proposal in smart-home environments. In this thesis, we question the conditions for a large-scale adoption of security solutions such as Intrusion Detection Systems (IDS), aiming at protecting constrained devices already deployed. Thus, a first contribution identifies the characteristics of smart homes to cross them with IDS taxonomies, in order to propose the qualitative criteria of a realistic domestic security solution.

Subsequently, in order to facilitate the design of IDS, a second contribution provides the scientific community with a Zigbee dataset, participating to the availability of tools covering the main protocols found in smart homes. All the frames exchanged by 10 devices during 10 days were captured by 4 probes distributed in a test house. Attacks have been introduced in order to establish and compare different detection strategies. In addition to MAC layer data redundancy, the dataset derives its originality from the extraction by each probe of the RSSI (Received Signal Strength Indicator) of each frame. This physical layer feature, accessible easily in most wireless technologies, allows to participate to the identification of fixed nodes. Later, one can imagine identifying each device more robustly by a physical layer fingerprint made of a tuple of several RSSIs, a complex combination to imitate by an attacker.

Finally, in a third contribution, we exploit the dataset to propose several IDSs detecting spoofing attacks, favored by the fact that several protocol stacks integrate little or no authentication on their MAC layer. To detect them, the consistency of the MAC layer identifier and the previous RSSI-based fingerprint can be considered, but this is no longer possible when the environments are constantly redrawn by the evolving inhabitants, as the RSSI becomes volatile. By providing RSSI time series as input to an unsupervised learning algorithm, we establish for each (device, probe) pair a model of normal RSSI sequences. Deviations from this model help detect an attack. The obtained detection metrics, which are very interesting given the low complexity of the initial considered architecture, as well as the evaluations of the autonomy and cost of the solution, suggest the spread of such systems in smart homes.

Keywords: Internet of Things security, Intrusion detection systems, Datasets, Spoofing attacks, Deep learning, Zigbee.

Acronymes

ANN	Artificial Neuron Network
API	Application Programming Interface
BLE	Bluetooth Low Energy
CDF	Cumulative Distribution Functions
COTS	Commercial Off-The-Shelf
DoS	Denial of Service
DDoS	Distributed Denial of Service
FCS	Frame Check Sequence
FFD	Full function Device
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
HIDS	Host-based IDS
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
IETF	Internet Engineering Task Force
IID	Independent and Identically Distributed
IoT	Internet of Things
IT	Information Technologies
ISM	Industrial Scientific Medical
LSTM	Long Short-Term Memory
MAC	Media Access Control
MAE	Mean Absolute Error
MSE	Mean Squared Error
NIDS	Network-based IDS
PSD	Power Spectral Density
PUF	Physical Unclonable Functions
RFD	Reduced Function Device
RNN	Recurrent Neural Network
RSSI	Received Signal Strength Indicator
SDR	Software-Defined Radio
WPAN	Wireless Personal Area Network
ZLL	Zigbee Light Link

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1 - Introduction

1.1 Contextes de travail et de recherche

1.1.1 Contexte de travail

Cette thèse s'est déroulée de septembre 2019 à décembre 2023 au sein du laboratoire CRIStAL (Centre de Recherche en Informatique, Signal et Automatique de Lille), unité mixte de recherche (UMR 9189) sous la tutelle du CNRS et de l'Université de Lille. La thèse s'est déroulée au sein de l'équipe 2XS (*eXtra Small eXtra Safe*), dirigée par le Professeur Gilles GRIMAUD et a été supervisée par le Maître de Conférences HDR Michaël HAUSPIE. L'équipe 2XS est hébergée à l'IRCICA[1], une unité d'appui et de recherche (UAR 3380) associant le CNRS et l'Université de Lille, dont l'originalité est d'accueillir des équipes de différents laboratoires afin de favoriser des axes de travail transversaux. Au niveau du financement de la thèse, j'ai continué à percevoir pendant quatre ans mon traitement de PRAG du département Génie Électrique et Informatique Industrielle de l'IUT de Lille tout en bénéficiant chaque année d'un aménagement de service, financé par l'Université de Lille, me permettant de n'effectuer que 192 heures de cours sur les 384 exigées normalement.

L'équipe 2XS se préoccupe de sécurité sur les systèmes informatiques, majoritairement contraints, à l'image des objets de l'IoT (*Internet of Things*). Elle travaille sur deux positionnements complémentaires de la chaîne de sécurité :

- Un axe de travail se soucie de sécurité en amont, c'est-à-dire dès la conception des systèmes. Il se préoccupe de la robustesse des systèmes informatiques contraints face aux attaques logicielles permettant des accès à des zones mémoires non autorisées. Pour cela, un noyau de système d'exploitation formellement prouvé assurant une isolation mémoire des applications a été défini. Son nom est « pip », celui-ci a fait l'objet de nombreuses publications et de plusieurs thèses à l'image de (Jomaa, 2018) et (Dejon, 2022).
- L'autre axe se préoccupe de sécurité en aval, c'est-à-dire quand le système informatique est en fonctionnement (*at runtime*). Ainsi, cet axe se consacre essentiellement à la détection d'intrusion dans les systèmes d'information. Une partie des travaux porte sur les centres de données ; une thèse CIFRE avec l'hébergeur OVH est en cours (Boin et al., 2022), faisant suite à une première sur le sujet (Riquet, 2015). L'autre partie se consacre aux systèmes contraints de l'IoT et c'est ici que je m'inscris. Ainsi, les travaux de thèse décrits dans (Helluy-Lafont, 2021) proposent un outil de capture par radio logicielle bien adapté à la diversité des protocoles sans-fil de l'IoT. Ces travaux ont permis par exemple d'identifier dix types de smartphones par l'extraction-analyse de caractéristiques physiques de leurs signaux Bluetooth. En outre, une implémentation *open source* de la couche physique du Bluetooth Classic a également été développée dans cette thèse, mettant à la disposition des chercheurs un outil contribuant à évaluer la sécurité de ce protocole. Une vulnérabilité Bluetooth sur certains microcontrôleurs a également été mise jour puis publiée dans la base de données CVE (*Common Vulnerabilities and Exposures*) du MITRE (Helluy-Lafont, 2019).

[1] Institut de Recherche sur les Composants logiciels et matériels pour l'Information et la Communication Avancée

1.1.2 Contexte de recherche : la sécurité des objets connectés

Le 1er novembre 2019, le magazine Newsweek faisait sa une (cf. Figure 1-1) avec un titre énigmatique « *Can you trust your toaster?* » (en français : « Pouvez-vous faire confiance à votre grille-pain ? »), suivie d'un épais dossier de sensibilisation aux risques de l'utilisation sans discernement des objets connectés domestiques. L'existence d'un tel article à destination du grand public était symptomatique d'un début de prise de conscience dans la société quant aux dangers de ces objets censés faciliter notre quotidien.

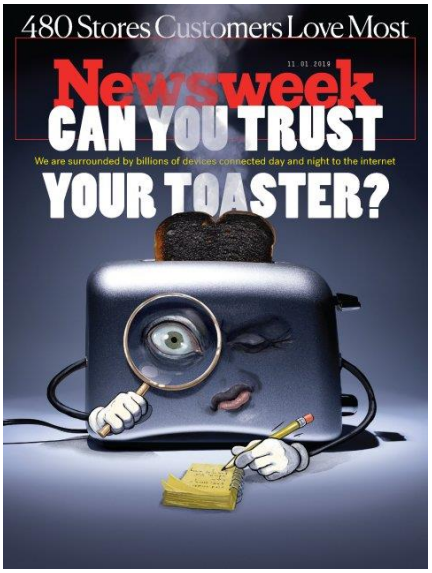

Figure 1-1. Une de l'hebdomadaire Newsweek du 1er novembre 2019

Reliant le monde physique au monde virtuel de la supervision, les objets connectés à la base de l'IoT ne cessent de se répandre dans tous les secteurs scientifiques et techniques : maison, santé, industrie, agriculture, transport, ville et réseaux d'énergie, prouvant le bienfondé de cette approche. En anglais, chacun de ces secteurs se voit alors qualifié de l'adjectif *smart* qu'on traduit en français par « intelligent » mais aussi plus sobrement par « connecté ». Ainsi, « maison connectée », « santé connectée », etc. sont devenues des expressions courantes. Dans le monde, le nombre d'objets impliqués est estimé à 22 milliards en 2025 (IoT Analytics, 2023) et une autre source (Usine Digitale, 2018) fait état pour la même année d'un chiffre d'affaires global de 1500 milliards de dollars. L'IoT a provoqué en ses quelques vingt-cinq années d'existence une révolution à la fois industrielle, sociétale et économique (ENISA, 2017).

Les objets connectés sont dotés de capacités de communication, souvent sans-fil, afin de les intégrer à des réseaux. La nature ouverte du medium utilisé menace la confidentialité, l'intégrité et la disponibilité des données ou des services davantage que les technologies filaires. Plus fragiles en termes de sécurité que les réseaux de l'IT (*Information Technologies*) classique, les réseaux intégrant des objets connectés peuvent être attaqués pour obtenir des informations relatives à la vie privée des utilisateurs ou pour réduire la fonctionnalité qu'ils sont censés offrir. En outre, une fois compromis, ils peuvent servir de porte d'entrée pour attaquer les réseaux domestique et professionnel dans lesquels ils s'insèrent. L'attaque de réseaux extérieurs aux nœuds compromis peut également être envisagée grâce à ces derniers. Les conséquences des attaques IoT sont souvent physiques et donc différentes de celles de l'informatique traditionnelle ; elles peuvent aller du simple dérangement lié au manque de réactivité d'une ampoule jusqu'à la mort d'un utilisateur via l'attaque par déni de service du système d'antiblocage des roues de son véhicule. L'éditeur

de solutions de sécurité informatique Kaspersky fait état au premier semestre 2021 de 1.5 milliards de cyberattaques visant des dispositifs IoT (Threatpost, 2021). L'impact économique et humain de celles-ci reste difficile à évaluer mais on cite des ordres de grandeur de plusieurs centaines de milliards de dollars de dégâts depuis les débuts de l'IoT (Radanliev et al., 2018). Dès que les objets connectés ont commencé à faire partie de la vie courante, les cybercriminels ont rapidement identifié et exploité les nouvelles opportunités qu'ils représentaient en termes d'attaques possibles.

En complément du développement d'objets sûrs (premier axe de 2XS), il s'avère donc nécessaire de protéger les objets et réseaux IoT existants (deuxième axe de 2XS). Une voie possible est celle de la synthèse de systèmes de détection d'intrusion (*Intrusion Detection Systems*, IDS) pouvant par exemple repérer des comportements anormaux dans les objets ou les réseaux. Mais derrière l'expression « objet connecté », se cachent un large spectre de réalités matérielles et logicielles, retardant le développement et la mise en œuvre de standards dans la sécurité IoT. Ainsi, en 2022, 43% des organisations utilisant de l'IoT ne sécurisaient pas complètement l'infrastructure en relation, entre autres à cause de la très grande diversité des objets (Kaspersky, 2022).

Dans cette thèse, je me suis intéressé spécifiquement au sous-domaine de l'IoT qu'est la maison connectée. Plus que les autres, ce secteur, destiné à des consommateurs, est très agressif sur le plan économique et n'a pas manqué d'attirer des fabricants peu au fait de la sécurité, surtout à l'affut d'opportunités financières. On peut par exemple acheter en ligne des ampoules Wi-Fi à LED pour moins de 5 € TTC l'unité, résultat d'une gestion exclusivement pilotée par le coût final. Celle-ci est souvent à l'origine d'un dimensionnement des ressources des objets au plus juste, reléguant la sécurité au second plan. En outre, sur un plan technologique, ce domaine de l'IoT expose plus que les autres une grande variété de piles de protocoles, ce qui a pour effet d'augmenter les surfaces d'attaque et par là de complexifier la détection d'intrusions. Enfin, sur un plan humain, les objets des maisons connectées sont « administrés » par des utilisateurs peu au fait des bonnes pratiques de sécurité qui ont chez eux un grand nombre de petits objets qui ne recevront de l'attention qu'au moment de leur installation. Toutes les conditions sont réunies pour faire de la maison connectée un repaire de vulnérabilités qui sont et seront exploitées les unes après les autres, compromettant potentiellement le rapport bénéfices / risque des objets, la confiance qu'ont les utilisateurs en ceux-ci et les opportunités de marché.

L'éducation à la cybersécurité de l'IoT et sa légifération font cela dit leur chemin. On peut citer pour commencer le projet en ligne OWASP (OWASP, 2018) qui recense une liste des mauvaises pratiques transformant les menaces en attaques. Aux États-Unis, l'organisation à but non lucratif MITRE maintient une base de données des vulnérabilités (MITRE, 2022) ; le Département de la sécurité intérieure (*U.S. Department of Homeland Security*) a édité un guide sur la conduite à tenir pour sécuriser l'IoT (Homeland Security, 2016) ; le NIST (*National Institute of Standards and Technology*), est passé à l'étape supérieure en rédigeant en 2020 une loi intitulée « *IoT Cybersecurity Improvement Act of 2020* » (Kelly, 2020). En Europe, l'ENISA (Agence Européenne pour la cybersécurité) publie des guides de recommandation notamment pour l'IoT (ENISA, 2017) et participe à la mise en œuvre du *Cyber Resilience Act* de 2022 (European Union, 2022). Cette loi sur la cyber-résilience fixe des règles de sécurité informatiques à respecter par les fabricants pour que leurs produits puissent recevoir le marquage CE, sésame pour investir le marché européen. À titre d'exemples, la cybersécurité doit être prise en compte dans toutes les phases du cycle de vie de l'objet, un objet doit pouvoir recevoir des mises à jour de sécurité pendant cinq ans et les vulnérabilités et incidents doivent être reportés.

L'IoT et particulièrement la maison connectée sont des domaines jeunes, peu matures et en constante évolution ; il va certainement se passer encore plusieurs années avant d'observer

les effets vertueux d'une sécurité abordée dès la conception et d'une législation exigeante. En attendant, un grand nombre d'objets peu sûrs sont déployés et en fonctionnement. Je pense qu'il y a une place pour des solutions de sécurité réalistes les concernant.

1.2 Questions de recherche

De nombreuses questions de recherche émergent à la suite de ces constats concernant la sécurité des maisons connectées. Je les livre ici sans hiérarchisation :

- Peut-on contribuer à sécuriser les petits objets « livrés à eux-mêmes » ?
- Si oui, contre quels types d'attaques peut-on espérer les protéger ?
- Quelles architectures et algorithmes les solutions de sécurité devraient-elles adopter pour espérer être diffusées dans le grand public ?
- Y'a-t-il des protocoles IoT plus envisagés que d'autres dans la proposition d'IDS ?
- Comment prendre en compte l'hétérogénéité des protocoles et celle des objets ?
- Quels outils d'évaluation seraient bénéfiques aux concepteurs de solutions de sécurité ?
- Si des solutions de sécurité sont envisagées, comment appréhender le facteur humain dans leur gestion ?
- Comment s'assurer que les identités annoncées par les objets sont véridiques ?

1.3 Plan du manuscrit et contributions

1.3.1 Plan du manuscrit

Au long de ce manuscrit, je tenterai de répondre aux différentes questions de recherche posées précédemment. Dans ce but, celui-ci est organisé de la façon suivante.

Dans le chapitre 2, je présente l'état de l'art relatif au contexte scientifique de la thèse : IoT, maison connectée, attaques, protocole d'étude retenu, taxonomies d'IDS et notion d'empreinte d'objet sont ainsi abordés.

Fort de l'état de l'art constitué, le chapitre 3 fait tendre les questions de recherche vers l'exposé de la problématique que je soulève dans mon travail. La thèse soutenue est ensuite exposée avant d'en détailler l'approche.

Le chapitre 4 présente une première contribution proposant une approche holistique du contexte « maison connectée » afin de définir une liste d'exigences pour des IDS bien adaptés à cet environnement. Après notamment un croisement de celles-ci avec des taxonomies d'IDS, des architectures et des algorithmes de détection cohérents avec le contexte sont proposés.

Le chapitre 5 décrit une deuxième contribution consistant en l'établissement d'un outil de type jeu de données pour le protocole domotique Zigbee. Il traduit tous les échanges entre 10 objets pendant 10 jours, captés par 4 sondes dans un domicile test habité. Des attaques ont également été insérées. Un exemple naïf d'IDS est proposé finalement pour une prise en main du jeu de données et une base de comparaison.

Utilisant le jeu de données précédent, une troisième contribution est décrite dans le chapitre 6. Celle-ci consiste en une détection des attaques d'usurpation d'identité dans les contextes habités, sans cesse redessinés.

Enfin, le chapitre 7 conclut cette thèse en résumant les contributions apportées. Il ouvre la discussion en proposant quelques perspectives.

1.3.2 Publications

Les trois contributions des chapitres 4, 5 et 6 de cette thèse ont fait l'objet de communications auprès de la communauté scientifique :

- Première contribution :
    - Journée thématique du 11 mai 2021 du GT SSLR : (Lourme and Hauspie, 2021a),
    - WiMob international conference 2021 : (Lourme and Hauspie, 2021b).
- Deuxième contribution :
    - WiMob international conference 2023 : (Lourme et al., 2023),
    - Jeu de données Zigbee 2023 : (Lourme and Hauspie, 2023a).
- Troisième contribution :
    - Conférence RESSI 2023 : (Lourme and Hauspie, 2023b), article et poster.

1.3.3 Médiation

Cette thèse comporte également un volet médiation par :

- la rédaction et la maintenance de l'article Wikipédia portant sur les réseaux étendus à basse consommation (LPWAN) : (Lourme, 2020a) ;
- la rédaction d'un article pour le mensuel Programmez ! : « ESP32, Mongoose OS et GCP Cloud IoT Core : un trio efficace et sûr pour l'IoT » (Lourme, 2020b).

1.3.4 Éducation

Enfin, cette thèse comporte un volet éducatif par la supervision du travail de deux étudiants de Master 2 lors de l'année 2022/2023 :

- le premier travail appelé « Thèse de Master » est de nature bibliographique et concerne un état de l'art de la sécurité dans l'IoT ;
- le second travail appelé « Projet de Fin d'Études » est un travail technique abordant en Python la gestion et l'exploitation d'un flux de trames réseaux, soit capturées en direct par une sonde, soit extraites d'un fichier PCAP.

Chapitre 2 - État de l'art

Dans ce chapitre, le contexte scientifique de la thèse est établi. Les caractéristiques des environnements IoT et les principales piles de protocoles sont d'abord présentées. Par la suite, le contexte particulier de la « maison connectée » et les défis de sécurité associés sont envisagés sous une triple perspective : économique, technologique et humaine. Un état de l'art des attaques est ensuite présenté afin de cerner celles auxquelles j'aimerais me consacrer. Le cas particulier du standard Zigbee qui sert de cas d'étude désormais est spécifiquement étudié, ainsi que ses vulnérabilités. Par la suite, un état de l'art des systèmes de détection d'intrusion est effectué. Enfin, les moyens d'identification radio sont présentés dans une dernière section qui résume les différentes techniques associées à la génération d'empreintes d'objets IoT et à leur utilisation.

2.1 Modèle « Internet of Things »

2.1.1 Piles de protocoles et objets contraints

2.1.1.1 Des Internets d'Objets plutôt qu'un Internet des Objets

L'expression « Internet of Things » (« Internet des Objets », en français) apparaît en 1999 (Ashton, 2009) et s'emploie désormais couramment, notamment via son acronyme « IoT ». Elle permet à l'époque de nommer un modèle émergent dans le traitement de l'information qui se distinguait du traditionnel « Internet des ordinateurs » circonscrit à un monde virtuel. Au contraire, l'Internet des Objets offrait (et offre toujours) un accès inédit au monde physique, laissant espérer une meilleure gestion des systèmes en favorisant par exemple des prises de décision rapide. Un exemple simple et classique est celui de la surveillance de la température d'une maison et l'action sur la consigne d'un thermostat pour ajuster celle-ci, le tout depuis une application installée sur un smartphone, présent dans la maison ou éloigné de celle-ci.

Cela dit, il est important de noter que, toujours aujourd'hui, l'expression « Internet des Objets » n'est pas une transposition de l'Internet des Ordinateurs à celui des Objets. Alors qu'il existe un internet des ordinateurs où potentiellement chacun d'entre eux peut se connecter à n'importe quel autre en s'appuyant sur la pile TCP/IP, les objets IoT sont des systèmes embarqués qui se sont vus adjoindre des possibilités de communication adaptées à leur contexte de travail, laissant prospérer une diversité de piles de protocoles dans l'IoT constituant des obstacles aux interconnexions d'objets. Au final, plutôt qu'à un Internet des Objets, on a à faire à un très grand nombre de petits Internets d'Objets à l'image de ceux de la maison de la Figure 2-4 page 36. Les réseaux avec une pile de protocoles différente de TCP/IP sont en général équipé d'une passerelle vers cette pile afin de communiquer avec l'extérieur. C'est le cas par exemple de « Smartphone 1 » et de « Passerelle Zigbee-Internet » sur la Figure 2-4. Sur cette même figure, il n'est cependant pas d'interopérabilité permettant à l'objet « Enceinte BLE » de dialoguer avec l'objet « Ampoule Zigbee 2 ».

2.1.1.2 Piles de protocoles des réseaux IoT

Les objets d'un réseau IoT parviennent à communiquer entre eux par le respect d'une technologie de communication commune. Celle-ci est implémentée sous la forme d'une pile de protocoles.

Par « réseau IoT », je me réfère à un ensemble d'objets reliés pour servir un propos commun. Ils observent la même pile de protocoles pour dialoguer entre eux et présentent notamment un système d'adressage cohérent.

En 2022, environ un tiers des objets intègre une pile liée à des technologies WLAN (*Wireless Local Area Network*), typiquement Wi-Fi, et un autre tiers intègre une pile liée à des technologies WPAN (*Wireless Personal Area Network*), typiquement Zigbee ou Bluetooth, (IoT Analytics, 2023).

Pile Wi-Fi – La pile Wi-Fi, dont une représentation est donnée Figure 2-1, est la plus répandue dans l'IoT car, tout d'abord, étant une instance du modèle TCP/IP, tout objet l'implémentant devient « naturellement » un nœud d'Internet. Cet objet est alors accessible théoriquement depuis tout autre nœud d'Internet, caractéristique allant dans le sens de l'esprit « supervision » de l'IoT. Ensuite, la plupart des environnements de déploiement intègrent déjà un point d'accès Wi-Fi, élément au centre de la topologie étoile par lequel les données des nœuds Wi-Fi transitent (je n'envisage pas dans mes travaux le cas des réseaux Wi-Fi *ad hoc*). Cela permet de ne pas investir dans des infrastructures tout en appliquant des règles de filtrage par l'entremise du pare-feu réseau implémenté dans le point d'accès. Un autre critère expliquant le succès de cette pile est que les ingénieurs et chercheurs de l'IT abordant des projets IoT sont habitués au modèle TCP/IP, à sa fiabilité et à sa possibilité de débits importants. Ils savent que, si besoin, TCP est un protocole de transport robuste et que l'ajout de TLS permet d'amener de la sécurité à une application HTTP. Ainsi, beaucoup d'articles avec le terme IoT dans leur titre ne traitent souvent et exclusivement que du Wi-Fi. Ce biais de surreprésentation se retrouve aussi dans la disponibilité des jeux de données. Pour revenir à des considérations techniques, l'exposition quasi directe à Internet peut représenter une menace si les solutions de sécurité sont mal configurées. Ainsi, les 1.5 milliards d'attaques IoT citées dans l'introduction (Threatpost, 2021) concernent surtout de l'IoT TCP/IP (des vulnérabilités sur les piles BLE et Zigbee sont régulièrement mises à jour mais il est difficile d'obtenir des chiffres d'attaques sur ces piles). En outre, les protocoles TCP/IP réclament des ressources car ils n'ont pas été conçus pour l'IoT et la « légèreté » de ses objets. Ainsi, les bureaux d'étude établissent des compromis en termes de sécurité et de coût en réalisant des adaptations des protocoles standards aux ressources disponibles des objets. Les auteurs de (Kolias et al., 2016) décrivent par exemple une implémentation de TLS maison sans vérification de certificat faute de ressources de chiffrement, conduisant à une attaque de type Man-in-the-Middle. Enfin, en termes énergétiques, eu égard aux portées et débits qu'elles permettent, les piles Wi-Fi consomment plus que les autres piles de protocoles évoquées ci-après.

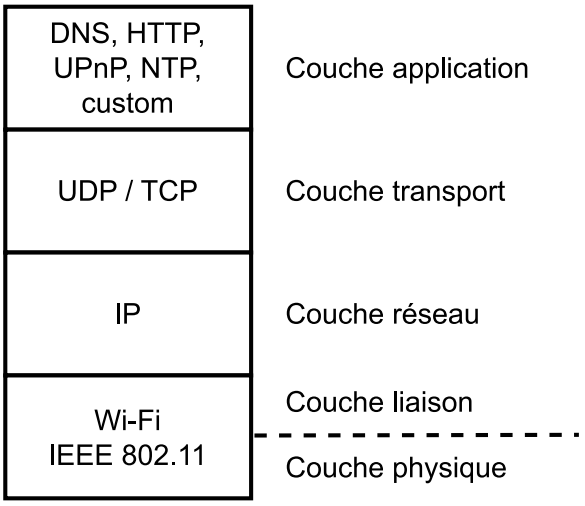

Figure 2-1. Pile Wi-Fi

Pile Zigbee – Le standard Zigbee a été conçu pour l'IoT (faibles ressources, faibles quantités de données). La possibilité de réseaux *ad hoc*, décentralisés, i.e., sans infrastructure, offre une résilience et des topologies (cf. Figure 2-8) que ne permettent pas les réseaux Wi-Fi, centralisés. Les consommations électriques sont optimisées et au final, il est moins cher de surveiller une application par Zigbee que par les autres technologies courte portée comme Wi-Fi et Bluetooth (Mordor Intelligence, 2022). Afin qu'ils communiquent avec l'extérieur, il faut équiper ces réseaux d'une passerelle Zigbee vers Internet. À l'opposé du Wi-Fi, elle ne peut jouer le rôle de pare-feu réseau vu les topologies permises mais son existence limite naturellement l'exposition à l'extérieur. Parfois, les fabricants proposent un accès à une interface de programmation d'application (*Application Programming Interface*, API) dont la sécurité reste à évaluer, afin de communiquer avec la passerelle sur son interface TCP/IP pour que celle-ci administre les objets depuis Internet ; le cas échéant, cela laisse envisager une communication de machine à machine. Les expériences menées pendant cette thèse ont été réalisées sur des réseaux Zigbee, aussi la section 2.4 propose une explication plus détaillée de ce standard.

Pile Bluetooth Low Energy (BLE) – Le BLE est une variante simplifiée du Bluetooth Classic, apparue avec la version 4 (2010) de ce protocole. Comme son nom l'indique, elle est orientée faible consommation et est donc particulièrement adaptée aux objets connectés. Son usage dans l'IoT est extrêmement répandu grâce à sa faible complexité et par le fait que tous les ordinateurs et smartphones vendus implémentent cette pile, jouant si nécessaire le rôle de passerelles vers Internet. BLE exploite la bande libre 2.4 GHz *Industrial Scientific Medical* (ISM) avec une modulation de type *Gaussian Frequency-Shift Keying* et un mécanisme de sauts de fréquence, limitant ainsi les interférences avec Wi-Fi et Zigbee qui utilisent également cette bande de fréquence la plupart du temps. La topologie est souvent de type point à point, même si les réseaux maillés sont possibles depuis Bluetooth 5.

Pile IETF pour l'IoT – Il est intéressant de mentionner ici le développement par l'IETF (*Internet Engineering Task Force*) d'une pile ouverte et sécurisée pour l'IoT, utilisant les réseaux Internet mais dédiées aux objets contraints en processeur, mémoire et énergie (Tschofenig and Baccelli, 2019). Les deux suites de protocoles utilisées, de la couche physique à la couche application, sont :

- IEEE 802.15.4 – 6LoWPAN/IPv6 – UDP – CoAP ;
- IEEE 802.15.4 – 6LoWPAN/IPv6 – TCP – MQTT.

6LoWPAN permet de relier les réseaux IEEE 802.15.4 de l'IoT aux réseaux IPv6 de l'IT. CoAP (*Constrained Application Protocol*) est un protocole *RESTful* et s'appuie sur UDP alors que MQTT est un protocole *Publish-Subscribe* s'appuyant sur TCP. Les deux sont bien adaptés à la gestion de petites quantités de données, cas fréquent dans l'IoT. Cette pile représente un espoir de standardisation dans l'IoT mais je ne suis pas parvenu à trouver des chiffres concernant sa réelle diffusion.

Synthèse sur les piles de protocoles – À travers l'exposé des principales piles de l'IoT, j'ai pu mettre en évidence une partie de l'hétérogénéité de l'IoT. Étant basée sur les protocoles de l'Internet, la pile Wi-Fi bénéficie d'une connectivité étendue naturelle, d'une diffusion commerciale importante et d'une couverture scientifique marquée. Cela dit, elle n'est sans doute pas optimisée pour les objets de l'IoT, notamment ceux à faibles ressources. Quant aux piles Zigbee et BLE, elles ont été optimisées pour les problématiques de ressources de l'IoT mais elles impliquent des passerelles qui malgré tout ferment les réseaux. Enfin, on peut signaler l'existence de travaux visant à modéliser les réseaux IoT hétérogènes sous une approche générique et abstraite. Je ne m'inscris pas dans cette voie mais elle mérite d'être mentionnée (Tournier, 2021).

2.1.1.3 Objets connectés contraints

Un microcontrôleur est un petit système informatique sur une seule puce (*System-on-Chip*, SoC) dotée d'un processeur, de mémoire et de périphériques d'entrées-sorties lui permettant d'interagir avec l'extérieur. Dans sa forme la plus simple, un objet, représenté Figure 2-2, est constitué en son cœur d'un microcontrôleur auquel sont reliés grâce à ses périphériques d'entrées-sorties des capteurs et des actionneurs. Le microcontrôleur est aussi relié à un circuit émetteur-récepteur sans-fil (*transceiver*) afin d'assurer à l'objet une capacité de communiquer. Cette partie radio est souvent la partie la plus énergivore d'un objet connecté. En utilisant un microcontrôleur, les facteurs SWaP-C[2] (taille, poids, consommation et coût) de l'objet connecté peuvent être optimisés pour sa tâche bien définie (Dejon, 2022). En général, un microcontrôleur est équipé d'un système d'exploitation, si possible à faible empreinte mémoire (Hahm et al., 2016), sur lequel les applications peuvent tourner. L'ensemble des programmes (système d'exploitation et applications) forment le micrologiciel (*firmware*).

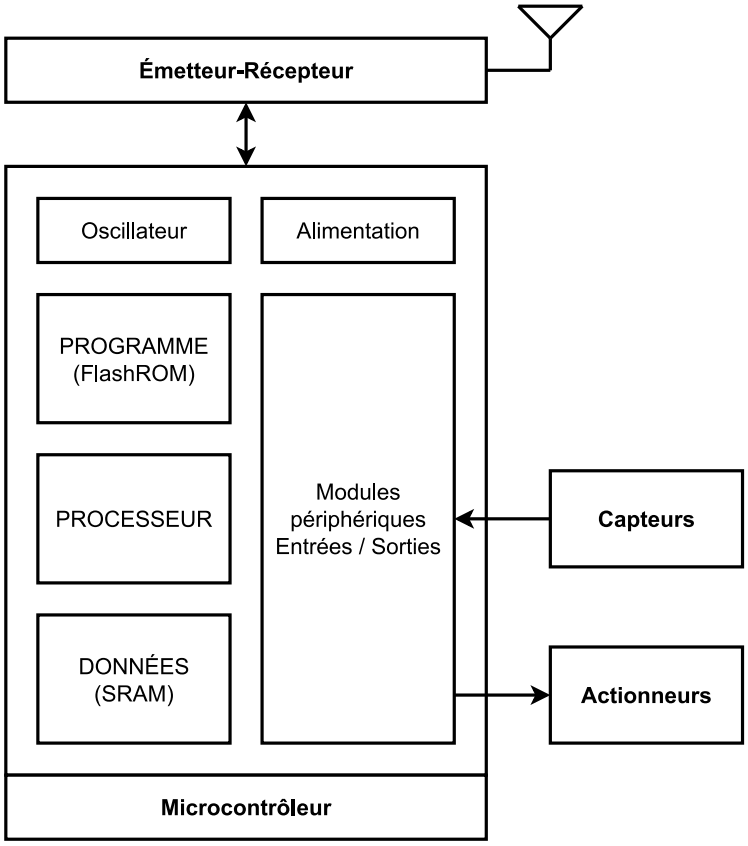

Figure 2-2. Constitution d'un objet connecté

Derrière cette description qualitative se cache une grande variété d'objets et donc de microcontrôleurs, allant de celui bas de gamme pilotant une cafetière Bluetooth à ceux hauts de gamme présents dans une télévision connectée. Dans le premier cas, on a à faire à ce qu'on appelle un objet contraint (*constrained device*) pour lequel les ressources (capacités de calcul et capacités de stockage) disponibles pour la tâche principale et des fonctions de sécurité (cryptographie par exemple) sont extrêmement réduites. Au contraire,

[2] https://nstxl.org/what-is-swap-c/

dans le deuxième cas, on trouve des microcontrôleurs dont les capacités peuvent être proches de celles des ordinateurs personnels et disposant d'une réserve de ressources conséquentes pour envisager des implémentations liées à la sécurité en plus des tâches principales.

Au sein des objets contraints, il existe même plusieurs degrés. L'IETF propose dans la RFC 7228 (Bormann et al., 2014) trois classes basées sur les volumes de mémoire vive et de mémoire morte, reportées Tableau 2-1. Dans cette thèse, je m'intéresse à la sécurisation de tous les objets contraints en commençant par les plus petits, c'est-à-dire ceux de la classe C0, estimant qu'ils auront probablement du mal à se défendre seuls vu leurs faibles ressources. Un nombre conséquent de ces objets sont déjà déployés dans les maisons. Avec le temps ils sont devenus vulnérables mais leurs micrologiciels ne feront jamais l'objet de mises à jour de sécurité soit parce que ce n'est pas prévu soit parce qu'il n'y a pas assez de ressources pour gérer ce processus.

Classe	**Taille des données (e.g., SRAM)**	**Taille du code (e.g., FlashROM)**	**Degré de contrainte**	**Exemple d'implémentation donné par la RFC**
C0	<< 10 Kio	<< 100 Kio	Très contraint	Pile d'un capteur dans un réseau de capteurs (Besoin de passerelle pour aller sur Internet)
C1	10 Kio	100 Kio	Assez contraint	Pile IETF pour l'IoT
C2	50 Kio	250 Kio	Moins contraint	Pile IETF pour l'IoT Pile Wi-Fi : début de considération

Tableau 2-1. Classes des objets contraints (IETF, RFC 7228)

Il faut noter que l'IETF donne ici des exemples d'implémentation en phase avec les standards qu'il promeut. À titre de comparaison, une ampoule couleur à LED Zigbee Philips Hue est aussi un objet contraint de classe 2. En effet, le microcontrôleur l'équipant est, en 2023, un Atmel SAM R21 (SRAM : 32 kio, FlashROM : 256 kio, oscillateur : 48 MHz) présentant une architecture 32 bits ARM Cortex M0+. Un objet implémentant une pile BLE se trouve également dans cette gamme de ressources.

L'implémentation d'une pile Wi-Fi dans un objet, même si elle dépend des protocoles effectivement présents dans la pile, ne peut en général se faire dans un appareil contraint de classe 2. En outre, concernant la ressource « énergie », des fonctionnements d'objets Wi-Fi sur batterie sont extrêmement rares eu égard à la consommation importante de cette technologie.

2.1.2 Vocabulaire de la cybersécurité, propriétés de sécurité

Dans ce manuscrit, des termes concernant la sécurité informatique seront souvent utilisés. Je donne ici les définitions des plus courants d'entre eux ; celles-ci sont la plupart du temps inspirées de (ANSSI, 2023).

La **cybersécurité** correspond à l'ensemble des moyens mis en œuvre pour qu'un système d'information garantisse la confidentialité, l'intégrité et la disponibilité des informations stockées, traitées ou transmises. Dans ce manuscrit, je substituerai régulièrement le terme « cybersécurité » par le terme « sécurité ».

Les propriétés de sécurité susmentionnées sont souvent présentées ensemble, formant la bien connue « triade de la sécurité ». Voici la définition de chacune de ces propriétés :

- **Confidentialité** : capacité à garantir que l'information n'est consultable que par les ayants droit.
- **Intégrité** : capacité à garantir que l'information n'est altérable que par les ayants droit.
- **Disponibilité** : capacité à garantir que l'information demeure accessible aux ayants droit.

L'**authentification** est le processus qui permet au système d'information de vérifier qu'une entité se présentant comme ayant droit d'accéder à l'information est bien celle qu'elle prétend être. L'authentification peut être basée sur ce que l'entité **est** (e.g., vérification d'une empreinte physique), ce que l'entité **possède** (e.g., une clé cryptographique), ce qu'elle **sait** (e.g., un mot de passe) ou ce qu'elle **sait faire** (e.g., un captcha).

De manière générale, une **cyberattaque** (ou **attaque**) représente tout acte malveillant envers un système d'information. Une **intrusion** est le fait pour des personnes, entités ou processus de pénétrer dans un espace (physique, logique, relationnel) où leur présence n'est pas souhaitée, c'est en général la première étape d'une attaque.

Les termes **faiblesse**, **vulnérabilité**, **menace** et **attaque** sont illustrés par le déroulé chronologique de la Figure 2-3. Celui-ci montre sur une « attaque de l'homme du milieu » (*Man-in-the-Middle attack*) comment le manque de ressource d'un objet peut conduire de proche en proche à une fuite d'informations confidentielles (Kolias et al., 2016).

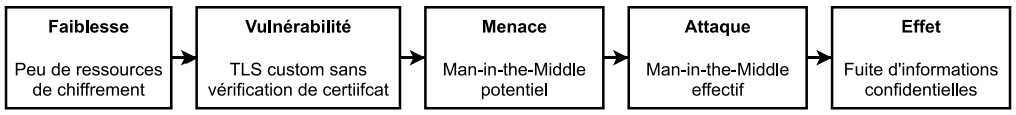

Figure 2-3. Quelques termes de cybersécurité

La **surface d'attaque** correspond à l'ensemble des éléments technologiques susceptibles de contenir une vulnérabilité exploitable par l'attaquant. Les **vecteurs d'attaque** relatifs décrivent les chemins par lesquels un attaquant peut réaliser les intrusions correspondantes. Le **modèle d'attaque** est un préalable à la conception d'une solution de sécurité. Définissant un cadre offensif, il fait des hypothèses sur les attaques que l'attaquant peut mener, les vecteurs d'attaques exploités et les ressources disponibles pour parvenir à ses fins. La solution de sécurité est évaluée dans ce cadre.

2.1.3 Défis de sécurité de l'IoT

Après avoir dressé un historique des objets connectés et présenté quelques aspects des piles de protocoles et des objets contraints, j'identifie ici les faiblesses de l'IoT en termes de sécurité.

2.1.3.1 Objets contraints en ressource

Essentiellement pour des critères économiques, les capacités de calcul et de stockage des microcontrôleurs utilisés dans l'IoT sont dimensionnées pour la mission principale de l'objet et permettent rarement d'envisager sereinement la sécurité des objets. Le temps d'exécution et l'empreinte mémoire nécessaires pour assurer via des moyens cryptographiques la confidentialité et l'intégrité des messages ou encore une authentification peuvent dans ce cas s'avérer rédhibitoires. De même, la fiabilité des communications peut souffrir de ressources insuffisantes provoquant congestions et collisions. Quant au débit de communication, quand il n'est pas contraint par des exigences de rapport cyclique, il reste de toutes façons très bas par rapport à une technologie filaire (typiquement quelques dizaines de kbits/seconde). Pour les objets fonctionnant sur batterie (télécommande, capteur de température, capteur de mouvement, etc.), l'énergie est également une ressource à gérer avec soin. Bons nombres d'objets sont prévus pour fonctionner dix ans avec leur pile. Les calculs, notamment cryptographiques, sont de gros consommateurs d'énergie, tout comme les communications radios. Ainsi, des modes « sommeil » consommant très peu sont souvent favorisés mais ils augmentent la latence du système. L'énergie est un vecteur d'attaque courant. Il est par exemple possible d'obtenir un déni de service (*Denial of Service*, DoS) par épuisement de la batterie en empêchant l'objet de passer en mode sommeil (*Denial of sleep*) par des sollicitations inutiles. Cette thèse n'aborde qu'anecdotiquement la sécurité sous l'angle de l'énergie.

2.1.3.2 Hétérogénéités des piles de protocoles et des matériels

Les piles de protocoles utilisés dans les réseaux IoT sont nombreuses et présentent donc de grandes diversités en termes d'ouvertures, portées, modulations, topologies réseau et menaces. Le problème de l'ouverture des piles est souvent sous-estimé : il est des piles où les implémentations de toutes les couches sont ouvertes et d'autres où certaines couches, propriétaires, sont exposées via des interfaces, contraignant le développeur à faire aveuglément confiance aux implémentations (pas toujours fiables) fournies. Les travaux décrit dans (Helluy-Lafont, 2021) parlent à ce sujet de l'« opacité des modems ». Par ailleurs, la diversité des microcontrôleurs et des systèmes d'exploitation disponibles (Hahm et al., 2016) conduit également à un nombre impressionnant de combinaisons. Globalement, ces manques de standardisation participent à ce que l'on peut appeler l'« opacité de l'IoT » (Siby et al., 2017) et ils dispersent les efforts de sécurisation.

2.1.3.3 Faiblesses inhérentes aux communications sans-fil

La plupart des objets connectés communique sans fil dans l'air. L'aspect ouvert de ce medium permet à un attaquant des actions à distance qu'il serait complexe à mener avec des technologies filaires : le brouillage (*jamming*), l'écoute passive (*eavesdropping*) et l'injection de messages (*message injection*). Ces attaques de communication sont décrites plus en détail dans la section 2.3, juste après la présentation du contexte « maison connectée ».

2.2 Contexte de la maison connectée

Cette section décrit d'abord la maison connectée, sous-secteur de l'IoT et cadre d'étude privilégié dans le cadre de cette thèse, puis elle aborde les défis de sécurité de ces environnements.

2.2.1 Écosystème de la maison connectée

Au sein de l'Internet des Objets (IoT), le secteur de la maison connectée est en plein essor. Pour quelques dizaines ou centaines d'Euros, ses habitants peuvent contrôler les appareils équipant leur domicile : électro-ménager, chauffage, lumières, lave-linge et systèmes d'alarme pour n'en citer que quelques-uns. La plupart du temps, cette gestion s'effectue via des applications installées sur un smartphone, depuis l'intérieur ou à distance de la maison. L'automatisation de fonctions domestiques par la programmation de scénarios procure un confort accru aux habitants. Particulièrement agressif sur le plan économique, le marché grand public de la maison connectée redessine complètement la manière de gérer les domiciles.

La Figure 2-4 est un exemple simple faisant figurer des objets utilisant différentes technologies de communication (Ethernet, BLE, Wi-Fi et Zigbee), chacune avec sa topologie. Ainsi, quand le propriétaire de « Smartphone 1 » envoie via son application mobile d'éclairage un ordre d'allumage à la lampe « Ampoule Zigbee 1 », le message relatif empreinte des réseaux Wi-Fi, Ethernet et Zigbee.

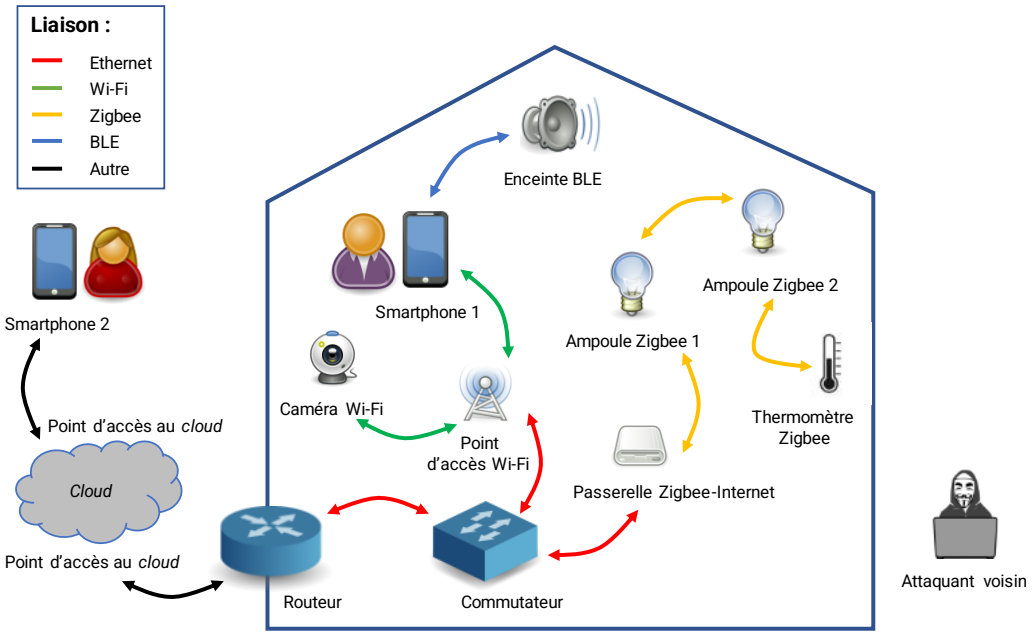

Figure 2-4. Exemple simple de maison connectée

Cette description est cela dit incomplète. Une partie de l'intelligence du système est maintenue dans des « nuages » (*clouds*), qui exposent des services via des points d'accès (*cloud endpoints*). Ces services consistent par exemple à administrer les objets à distance, générer des notifications, stocker les historiques de retours d'états et de commandes, mémoriser des préférences ou encore calculer des statistiques sur les habitudes de l'utilisateur pour lui faire des suggestions, etc. Cela ne va pas sans poser des questions de vie privée (surtout que le *cloud provider* est souvent une tierce-partie) mais l'utilisateur accepte la plupart du temps ces indiscrétions en validant les conditions générales lorsqu'il installe l'application mobile. Les *cloud endpoints* constituent aussi des points d'entrée vers la maison lorsque l'utilisateur est à l'extérieur de celle-ci.

Ainsi, quand ils évaluent la sécurité de 45 objets domestiques du marché, les auteurs de (Alrawi et al., 2019) considèrent à chaque fois les quatre surfaces d'attaque suivantes :

- l'objet lui-même ;
- l'application mobile interagissant avec l'objet et les *clouds* ;
- les points d'accès aux *clouds*, interfaces permettant d'accéder aux service Internet avec lesquels l'objet et l'application mobile communiquent ;
- les communications.

Cette approche quasi-exhaustive est intéressante. Néanmoins, le modèle d'attaque souffre d'être limité à un attaquant utilisant les réseaux IP. Cela ne pose pas de problème pour les objets Wi-Fi mais par contre les réseaux de type Zigbee ou BLE sont vus via leur passerelle IP, ce qui ne permet pas d'étudier les spécificités de ces réseaux en termes de sécurité, notamment celles liées aux communications sans-fil.

2.2.2 Défis de sécurité de la maison connectée : une approche holistique

Le domaine de la maison connectée hérite à quelques variations près des défis de sécurité de l'IoT citées dans la section 2.1.3 et il en ajoute un autre, lié au profil particulier de l'utilisateur. Afin d'appréhender la maison connectée autrement que sous un angle purement technique, je propose de réorganiser cette vision selon une approche holistique faisant intervenir trois facteurs : économique, technologique et humain. Je pense que cette vision originale dans la littérature constitue une base intéressante pour la synthèse d'IDS réalistes et par-là adoptés.

2.2.2.1 Facteur économique

Un marché piloté par le coût, induisant des objets contraints mal sécurisés – Par rapport aux autres secteurs de l'IoT qui sont à destination des professionnels (santé ou industrie par exemple), le marché de la maison connectée est essentiellement piloté par le coût et le *time-to-market*. Ainsi, conception, fabrication, diffusion, exploitation et recyclage doivent coûter le moins cher possible au fabricant afin de parvenir à un prix de vente qui déclenche l'acte d'achat chez le consommateur envieux de nouvelles fonctionnalités à son domicile. Par exemple, une ampoule Zigbee coûte selon sa qualité entre 5 et 20 €. Ces perspectives de marché amènent différents biais préjudiciables à la sécurité. Tout d'abord, beaucoup de fabricants d'objets ne viennent pas du monde de l'informatique traditionnelle ; ils ont souvent de ce fait une faible culture de la sécurité et une expertise basse dans le développement de micrologiciels (*firmwares*) sûrs. Certains fabriquaient par exemple un certain type d'objet qu'ils ont décidé de rendre « connecté », en assemblant des composants « sur étagère » (*Commercial Off-The-Shelf*, COTS) à la sécurité globale douteuse. Par ailleurs, la possibilité de mise à jour de l'objet[3], pour des améliorations ou pour appliquer un patch correctif, est rarement envisagée, soit par manque de ressources disponibles, les microcontrôleurs retenus étant souvent contraints, soit pour des raisons économiques ou par peur de casser la rétrocompatibilité. Au final, on se retrouve souvent avec des objets à ressources minimales, livrés à eux-mêmes, avec des implémentations de sécurité dégradées, conduisant par exemple à des communications sans-fil ne respectant pas la confidentialité ou des pratiques d'authentification ne respectant pas correctement les standards (Kolias et al., 2016).

[3] On peut sur le sujet suivre les travaux du groupe de travail *Software Update for Internet of Things* (SUIT) de l'IETF.

2.2.2.2 Facteur technologique

Plusieurs technologies de communication dans un environnement dynamique – Dans le jeune domaine de la maison connectée, le nombre de standards disponibles est conséquent : Wi-Fi, BLE, Zigbee pour les plus courants mais aussi Z-Wave, EnOcean, Enhanced ShockBurst, 6LoWPAN, LoRaWAN, etc. La multiplication des standards multiplie les surfaces d'attaques et donc les études de solutions de sécurité. Par ailleurs, les environnements de type maison connectées sont dynamiques à double titre. D'abord, des objets sont ajoutés et retirés régulièrement dans l'environnement. Ensuite, par leurs mouvements dans le domicile, les meubles qu'ils déplacent, les portes qu'ils ouvrent et ferment, les utilisateurs redessinent en permanence l'environnement dans lequel évoluent les signaux émis et reçus par les objets, ce qui influence leurs caractéristiques électromagnétiques. Le cas des objets mobiles n'est pas considéré dans ce manuscrit.

2.2.2.3 Facteur humain

Un utilisateur profane au milieu d'objets déjà déployés – À l'opposé d'un contexte industriel, les utilisateurs de maisons connectées ne sont généralement pas techniciens, ingénieurs ou administrateurs réseau, ils souhaitent juste profiter de leur installation avec le minimum d'ennui et d'engagement dans un esprit *setup and forget* où la maintenance est exclue. La plupart du temps, ils ont une faible culture de la sécurité et en sont logiquement les maillons faibles. En 2010, une analyse à très large échelle[4] (Cui and Stolfo, 2010) a révélé que 13% des 500.000 appareils découverts sur le net avaient conservé leur mot de passe super-utilisateur par défaut. De même, les utilisateurs sont en général peu regardants sur le respect de leur vie privée, acceptant moyennant de nouvelles fonctionnalités de laisser filer des informations sensibles vers les *clouds* (Ren et al., 2019) ; dans le même registre, ils n'ont pas conscience que leurs réseaux IoT peuvent être écoutés passivement, prélude à des actions plus offensives. En outre, les objets n'ayant pas d'interface homme-machine, le seul moyen d'interagir avec eux est l'application mobile ; il est peu probable que celle-ci soit conçue pour remonter des attaques. Ainsi, celles-ci peuvent rester longtemps non remarquées alors que plus elles sont détectées tôt, plus limités sont les dommages. Également, quelle mitigation peut être envisagée quand une attaque est détectée ? Envoyer une notification à l'utilisateur pour qu'il débranche l'appareil attaqué ou tout le système ? Tout dépend de ce que la détection est capable de déterminer.

2.3 Attaques IoT : une taxonomie

En corollaire à ce déploiement massif d'objets connectés fragiles, de nombreuses attaques ont lieu, conduisant à des ruptures de confidentialité, d'intégrité et de disponibilité. Ainsi, nous avons tous entendu parler de fuites de données privées permettant de connaître les habitudes des utilisateurs, de prises de contrôle massives d'objets conduisant à des dénis de service distribués sur de grands acteurs du web, ou encore de déni de service, par exemple lorsqu'une ampoule ne répond plus aux ordres du smartphone associé.

Il est intéressant de noter que dans les trois exemples ci-dessus, les attaques sont souvent composées dans des scénarios pour parvenir à un but final où elles n'ont été qu'un maillon. Par exemple dans le premier cas, l'attaquant peut inférer la présence effective de l'utilisateur à son domicile dans le but d'entrer ensuite par effraction pendant son absence. Dans le second cas, il s'agit d'abord de prendre le contrôle du plus grand nombre d'objets possible dans le but de lancer à une date précise vers un site une attaque distribuée. Enfin, dans le troisième cas, la paralysie d'une ampoule peut être une nuisance suffisante en soi mais ce peut être aussi l'occasion d'inciter l'utilisateur à réassocier son ampoule au réseau, une phase pendant laquelle une clef cryptographique est échangée, parfois avec peu de

[4] À l'image de ce que permet https://www.shodan.io/

protection. Une taxonomie d'attaques Zigbee organisées en scénarios est présentée dans (Sadikin et al., 2020).

Pour détecter et contrer les attaques, il est nécessaire de les connaitre et donc de les catégoriser. Par-là, je serai en mesure de préciser celles auxquelles il me parait le plus « rentable » et « urgent » de me consacrer.

Précisons tout de suite qu'en relation avec les différentes surfaces d'attaques présentées section 2.2.1, je ne m'intéresse pas aux attaques sur l'application mobile (permises par exemple par des permissions excessives) ni aux attaques sur les points d'accès aux *clouds* (permises par exemple par des API tierces-parties vulnérables). Ainsi, je choisis de m'attarder seulement sur les attaques visant les objets eux-mêmes et leurs communications.

La taxonomie originale proposée par (Ronen and Shamir, 2016) classent les attaques selon leur rapport à la fonctionnalité d'origine de l'objet : ignorance, réduction, usage inapproprié ou extension de celle-ci. Bien qu'intéressante, je préfère la taxonomie proposée par (Tschofenig and Baccelli, 2019) ou par le MOOC sur les objets connectés que j'ai suivi au début de la thèse (INRIA, 2020) car elle s'adapte bien au cadre que je viens de définir. Reproduite Figure 2-5, elle classe les attaques retenues en trois catégories : de communication, logicielle et matérielle et les positionne selon la difficulté à les mener et le coût pour s'en protéger.

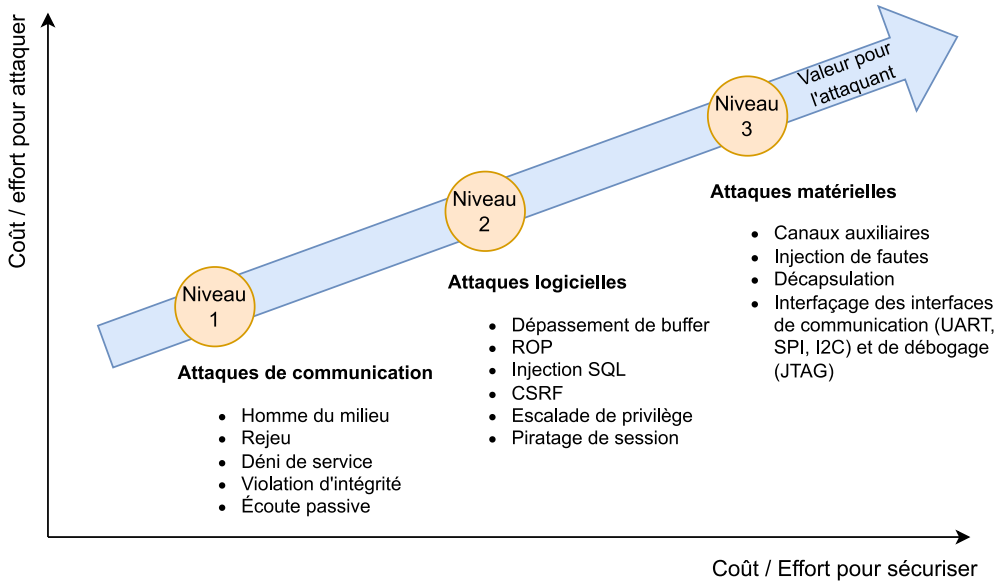

Figure 2-5. Taxonomie d'attaques retenue. SQL : Structured Query Language ; ROP : Return-Oriented Programming, CSRF : cross-site request forgery ; les autres acronymes sont définis dans le texte.

2.3.1 Attaques matérielles

Les attaques matérielles exploitent une vulnérabilité en lien avec des caractéristiques physiques de l'objet.

Un type d'attaque matérielle est celui dit « par canaux auxiliaires » (*side channel*). Il est par exemple envisageable d'obtenir une clé de chiffrement par la mesure de la consommation de courant de la puce réalisant ce chiffrement. C'est ce qu'on fait les auteurs

de (Ronen et al., 2017) sur une ampoule Zigbee. D'autres attaques par canaux auxiliaires exploitent aussi des mesures temporelles ou électromagnétiques.

L'attaque par injection de faute consiste, quant à elle, à envoyer des impulsions électromagnétiques de forte puissance sur l'objet pour que certaines fonctions critiques soient ignorées, permettant par exemple à l'attaquant de se retrouver à des séquences du programme où il n'aurait pas pu accéder normalement.

On peut aussi chercher à obtenir ou envoyer des informations par les canaux de communication filaires attachés à l'objet comme les liaisons série UART (*Universal Asynchronous Receiver-Transmitter*), SPI (*Serial Peripheral Interface*) et I2C (*Inter-Integrated Circuit*) ou l'interface JTAG (*Joint Test Action Group*) de débogage. Le but recherché peut être l'extraction d'une clé de chiffrement préinstallée ou un passage à un mode privilégié du microcontrôleur de l'objet pour injecter un code malicieux (*malware*).

En général, les attaques matérielles nécessitent une proximité physique avec l'objet attaqué, du matériel spécialisé onéreux et beaucoup de travail d'ingénierie. Même si elles sont particulièrement dangereuses, elles ne sont pas économiquement rentables pour un attaquant et il est peu probable qu'elles soient pratiquées dans le contexte à enjeu *a priori* non critique qui m'intéresse, c'est pourquoi je ne les considèrerai pas d'avantage.

2.3.2 Attaques logicielles

Injecter du code malicieux est également possible en exploitant des vulnérabilités dans le micrologiciel des objets. Par exemple, un micrologiciel mal conçu donnera accès, lors d'un dépassement de tampon (*buffer overflow*) provoqué par l'attaquant à des zones mémoire normalement inaccessibles. L'attaquant peut alors étudier l'ingénierie de l'objet, puis en prendre le contrôle en y injectant un *malware*, pouvant, selon sa sophistication, aller jusqu'à corrompre tout le réseau IoT. Il peut aussi s'organiser pour que l'objet fasse fuiter des informations sensibles vers l'extérieur.

Une fois le *modus operandi* d'une attaque de ce type établi, sa diffusion à large échelle est une sérieuse menace étant donné la nature connectée des environnements. On peut par exemple craindre une armée de *botnets* mettant en marche à pleine puissance tous les climatiseurs d'un quartier pour faire tomber le réseau d'énergie.

Pour ne pas en arriver là, il est intéressant de considérer entre le matériel et le système d'exploitation une solution de sécurité présentant des garanties en termes d'isolation mémoire, par l'usage de méthodes formelles par exemple. Dans cet esprit, l'équipe 2XS a développé dans une thèse récente un noyau nommé Pip-MPU, dédié aux objets contraints (Dejon, 2022). Un ingénieur de l'équipe a validé ce noyau sur RIOT, un des systèmes d'exploitation répandus de l'IoT. Je suis convaincu de l'intérêt de cette approche mais l'intégration d'un tel dispositif ne peut se faire qu'en amont, c'est-à-dire dès le début de la conception de l'objet et les fabricants doivent y être impliqués. En attendant que de telles pratiques soient généralisées, j'ambitionne de protéger les objets sans mécanisme de défense, déjà déployés dans les environnements connectés.

Les efforts nécessaires pour mener des attaques logicielles, même s'ils sont moindres que ceux liés aux attaques matérielles, restent relativement importants et les solutions de sécurité sont à développer au cas par cas, c'est pourquoi je n'envisage pas explicitement ce type d'attaque dans mes travaux.

2.3.3 Attaques de communication

Ces attaques s'appellent aussi parfois « attaques réseau ». Elles consistent à utiliser par l'attaquant la surface d'attaque que constituent les communications au sein d'un réseau IoT, le plus souvent sans-fil et présentant donc un medium ouvert. Je ne considère pas ici

les communications mettant en jeu l'application mobile ou les points d'accès aux *clouds*. J'envisage ici les attaques les plus courantes des quatre couches de la pile Zigbee (physique, MAC, réseau/routage et application) mais les illustrations se font avec Zigbee, BLE et Wi-Fi.

2.3.3.1 Attaques de couche physique

Au niveau de la couche physique, deux groupes d'attaque sont possibles : le brouillage et l'écoute passive.

Brouillage – Cette action sur la couche physique consiste pour un attaquant à envoyer un signal radio assez puissant pour occuper le medium ou y corrompre les données y circulant. Le brouillage peut être de type continu, aléatoire, réactif ou sélectif. La perte d'intégrité est une conséquence logique de cette attaque mais elle se traduit aussi souvent par un épuisement des ressources, notamment énergétiques : en effet, l'objet victime ne recevant pas de signal d'acquittement aux messages qu'il émet peut s'épuiser à les retransmettre. Dans les deux cas, cela peut conduire à un DoS. J'ai pu mettre en œuvre un brouillage sur un réseau Zigbee en émettant à 70 centimètres de sa passerelle une sinusoïde à la fréquence de celle du canal utilisé, ceci avec une simple radio logicielle (*Software Defined Radio*, SDR) BladeRF 2.0 de puissance de sortie 6 mW[5]. L'intensité de la puissance décroyant à l'inverse de la distance au carré, il faut rapidement un brouilleur de forte puissance pour perturber le réseau sans lui être « collé » mais dans ce cas, l'attaquant devient plus repérable. Pour montrer à un superviseur qu'il n'est pas victime de brouillage et reste disponible, un objet peut régulièrement émettre un signal de type *heartbeat*.

Écoute passive – L'écoute passive se joue au niveau de la couche physique où il s'agit de capter les ondes électromagnétiques et d'en extraire les informations qu'elles contiennent, sans en émettre. L'attaquant peut utiliser une SDR pour récupérer des données brutes de type (I, Q) et les exploiter ou choisir une sonde en écoute passive (*sniffer*) dédié à une technologie, par exemple un CC2531 pour Zigbee, et récupérer directement des données démodulées organisées en trames exploitables par le logiciel d'analyse Wireshark. L'écoute passive est difficilement détectable et peu de protocoles peuvent s'en protéger, aussi il est nécessaire d'être vigilant sur la confidentialité des données en transit. Mais même avec du chiffrement, les métadonnées des trames (longueur, débit, etc.) sont exploitables pour obtenir des informations sur la vie privée des utilisateurs d'une autre nature, plus intimes, que celles auxquelles nous sommes habitués avec l'informatique traditionnelle. Par exemple, les trames IEEE 802.15.4 issues d'un capteur de mouvement Zigbee Philips Hue transportent les informations de couches réseau et application chiffrées. Cela dit, j'ai établi que les trames les transportant avaient des longueurs déterministes, fonction de la détection effective d'un mouvement. Ce type de déterminisme est courant dans l'IoT. Cette caractéristique peut être facilement exploitée par un voleur avant de s'introduire dans une maison. Des travaux similaires sur la vie privée, basés sur les informations fournies par les débits de données d'objets Wi-Fi, sont décrits dans (Apthorpe et al., 2017). Dans (Acar et al., 2020), des mitigations sur ces fuites d'informations liées à la vie privée sont en plus proposées en injectant du faux trafic permettant de faire croire à un attaquant que l'utilisateur est présent alors qu'il ne l'est pas. Présentée ici comme un vecteur d'atteinte à la vie privée, l'écoute passive peut aussi jouer un rôle positif en fournissant les données recueillies à des IDS qui seront en mesure grâce à elles de déterminer si une intrusion est en cours.

[5] Le logiciel utilisé *osmocom_siggen* fait partie du cadriciel Osmocom Gnu Radio dédié au développement de projets pour les SDR.

2.3.3.2 Attaques de couche MAC

Je considère maintenant les attaques de couche MAC. Cette couche basse est sans doute une des plus vulnérables car :

- il arrive souvent qu'elle ne soit pas chiffrée ou alors seulement partiellement. C'est le cas respectivement de IEEE 802.15.4 dans la pile Zigbee ou bien encore du Wi-Fi jusqu'à 802.11w où seules les trames de données sont protégées mais pas celles de contrôle ni de gestion ;
- il arrive souvent que des actions, dont certaines sont sensibles en termes de sécurité, soient confiées à cette couche sans nécessité d'authentification cryptographique, par exemple la requête de données dans IEEE 802.15.4 et celle de dé-authentification en Wi-Fi.

Ainsi je liste les attaques suivantes sur cette couche :

Usurpation d'identité (*spoofing*) – Un attaquant peut forger une trame où il utilise l'identité d'un nœud légitime pour envoyer ou obtenir de fausses informations à un nœud faisant confiance à ce nœud légitime. Il peut aussi par exemple forger une trame de dé-authentification Wi-Fi vers le point d'accès Wi-Fi, avec comme paramètre l'adresse MAC d'un nœud qu'il veut voir exclu du réseau.

Les attaques de *spoofing* font souvent partie de scénarios d'attaque, et, en ce sens, elles sont particulièrement dangereuses. Par exemple, une fois déconnecté, l'utilisateur va probablement tenter de se connecter à un autre point d'accès, malveillant cette fois.

Il apparait donc très important d'être capable de détecter les tentatives d'usurpation dans un contexte où il n'y a pas d'authentification (cryptographique ou autre, cf. 2.1.2) ou bien lorsque celle-ci peut être facilement mise à mal.

Les attaques d'usurpation de type **mascarade** correspondent à un attaquant qui prend l'identifiant d'un nœud légitime (un seul identifiant, plusieurs nœuds physiques) alors que les attaques d'usurpation de type **sybil** correspondent à un attaquant prenant plusieurs identifiants (plusieurs identifiants, un seul nœud physique). Des exemples de telles attaques sont données en 2.4.3.1.

Inondation (*flooding*) – Les requêtes usurpées, comme tout autre requête, peuvent être pratiquées dans un mode de rythme élevé pour solliciter exagérément un nœud victime. Cette inondation de requêtes est le moyen le plus simple de provoquer un épuisement des ressources voire un déni de service. Par exemple, un attaquant peut envoyer des centaines de fois par minute une requête de données en apparence légitime à une ampoule ; celle-ci ne tardera pas à manquer de réactivité.

Attaque de l'homme du milieu (*Man-in-the-Middle attack*) – Les faiblesses précitées de la couche MAC facilitent ce type d'attaque où les objets A et B croient communiquer ensemble alors que l'attaquant M intercepte, modifie et relaie les informations entre A et B. Un exemple de mise en œuvre de cette attaque et de sa détection sont décrites pour un contexte BLE dans (Lahmadi et al., 2020).

2.3.3.3 Attaques de couche réseau/routage

Attaque par rejeu (*replay attack*) – Une attaque par rejeu consiste à rejouer une séquence de messages préalablement capturée et contenant des commandes ou des demandes d'information que l'attaquant veut réexécuter ou reformuler. Des mécanismes anti-rejeu basés sur un compteur s'incrémentant à chaque transmission sont présents dans les objets conçus avec soin mais les systèmes très bon marché (par exemple : ensemble télécommande/prise 433 MHz) en sont dépourvus, j'ai pu le vérifier.

Attaque sinkhole – Cette attaque de routage consiste pour un nœud malveillant à attirer vers lui le trafic environnant en annonçant de très bonnes métriques de routage. L'attaquant peut alors étudier le trafic depuis ce nœud central mais aussi l'arrêter.

2.3.3.4 Attaques de couche applicative

Attaque par force brute (*brute force attack*) – En général, les accès à la couche applicative sont conditionnés à la possession de clés cryptographiques (venant éventuellement de couche inférieures) ou d'un couple identifiant/mot de passe valide. L'attaque par force brute utilise un dictionnaire de couples identifiant/mot de passe et tente un accès avec chacun jusqu'à ce que celui-ci soit autorisé. En 2016, le ver Mirai a utilisé dans un premier temps cette technique pour accéder à l'application Telnet de milliers de routeurs et webcams dont les identifiants/mots de passe avaient été laissés à ceux par défaut (Kolias et al., 2017). Ensuite, il a injecté dans chaque objet compromis un *malware* dont le but était d'attaquer par DoS à une date commune des services majeurs du web. L'ensemble de ces attaques synchronisées a constitué un déni de service distribué qui a particulièrement marqué les esprits par son ampleur. L'attaque par force brute est une attaque favorisée par la négligence des utilisateurs à changer leurs mots de passe. Une éducation aux bonnes pratiques de sécurité permettra de diminuer la fréquence de ces attaques.

2.3.4 Redéfinition des attaques par l'apprentissage automatique

J'ai tenté de présenter de façon classique les attaques les plus courantes de l'IoT. Il faut être conscient que, depuis une dizaine d'années, le déploiement de l'intelligence artificielle atteint aussi la définition et la mise en œuvre des attaques contre l'IoT. Par exemple, le brouillage constant évoqué en section 2.3.3.1 devient inadapté. Il est de plus en plus remplacé par des brouillages basés sur l'apprentissage automatique, à la fois efficaces dans leur capacité de nuire, peu repérables et économes en énergie. Un autre exemple est celui des attaques contradictoires (*adversarial attacks*) où ici l'attaquant ne cherche pas à attaquer le système IoT lui-même mais l'IDS qui le protège. L'ajout de petites perturbations bien choisies en entrée de l'algorithme de l'IDS conduit à la dégradation de ses métriques de classification et rend, de fait, son intérêt limité (Bout et al., 2022). Ainsi, les attaques changent de nature continument et il n'est pas de taxonomie les concernant gravée dans le marbre.

2.3.5 Attaques considérées et pile d'étude retenue

Cet exposé des attaques a permis de cerner les menaces qui pèsent sur les réseaux IoT et leurs environnements. Écartant, pour ne pas me disperser, les surfaces d'attaques liées aux applications mobiles et celles liées aux points d'accès aux *clouds*, il me restait à étudier les menaces sur les objets et leurs communications. Les attaques matérielles et logicielles sur les objets sont à considérer mais je préfère m'intéresser aux attaques concernant les communications sans-fil au sein des réseaux d'objets contraints. J'imagine comme sur la Figure 2-4 page 36 un attaquant avoisinant la maison et qui par des moyens bon marché peut pratiquer les attaques de communication que j'ai décrites. Parmi celles-ci, les attaques d'usurpation d'identité retiennent le plus mon attention. Dans certains contextes que j'ai évoqués, elles sont très faciles à réaliser mais conduisent néanmoins à de grandes compromissions. Je suis intéressé par la recherche de moyens permettant d'authentifier les objets alors qu'il n'est pas prévu d'authentification cryptographique ou que celle-ci n'est pas viable. Dans la section suivante de l'état de l'art, je présente le standard Zigbee qui mérite que l'on s'intéresse à ces problématiques et qui nous servira dorénavant de cas d'étude. Je garde cependant toujours à l'esprit que les solutions de protection qui seront développées dans ce manuscrit devront être aisément transposables aux technologies Wi-Fi

et BLE afin de respecter l'hétérogénéité des piles de protocoles utilisées dans les maisons connectées.

2.4 Cas d'étude : Profil Zigbee Light Link

Cette section décrit d'abord de façon générale le standard Zigbee, très utilisé dans le cadre de la maison connectée. Il va m'accompagner pour la suite de mes travaux, même si je reviendrai régulièrement sur les autres protocoles IoT. Ensuite, la pile de ce standard est présentée en détail. Enfin, l'exposé de vulnérabilités connues est réalisé dans l'idée de les exploiter ultérieurement lors de la définition et de l'évaluation d'une solution de sécurité.

2.4.1 Présentation du standard Zigbee

Apparu en 2005, Zigbee est un standard pour l'IoT développé par la *Connectivity Standard Alliance*, faisant figurer des fabricants comme Samsung, Philips-Signify, Texas Instruments ou Ikea. Son but est de fournir une communication bidirectionnelle sans-fil fiable à des objets faible consommation faible coût s'insérant dans des WPAN, dont la couverture est typiquement inférieure à 100 mètres. La plupart du temps, un réseau Zigbee opère dans la bande libre ISM 2.4 GHz, disponible dans le monde entier, sur un des 16 canaux de 2 MHz de large, numérotés de 11 à 26. Les fréquences centrales successives de ceux-ci sont séparées de 5 MHz, si bien que les canaux ne se chevauchent pas. Le premier (n°11) est à la fréquence centrale 2.405 GHz, le second est à 2.410 GHz, etc. et le dernier (n°26) est à 2.480 GHz. Dans la pratique, les canaux n°15, 20, 25 et 26 sont les plus utilisés car ils sont à l'abri des interférences dues aux très larges canaux (22 MHz) n°1, 6 et 11 du Wi-Fi (cf. Figure 2-6 empruntée à (La et al., 2018)). Dans cette bande de fréquence 2.4 GHz, la modulation est de type O-QPSK (*Offset-Quadrature Phase Shift Keying*) et le débit théorique maximum est de 250 kbit/s. Un étalement de spectre à séquence directe (*Direct-Sequence Spread Spectrum*, DSSS) est utilisé pour améliorer la sensibilité du receveur, accroître la résistance au brouillage et réduire les effets des chemins multiples (*multipath*) des ondes.

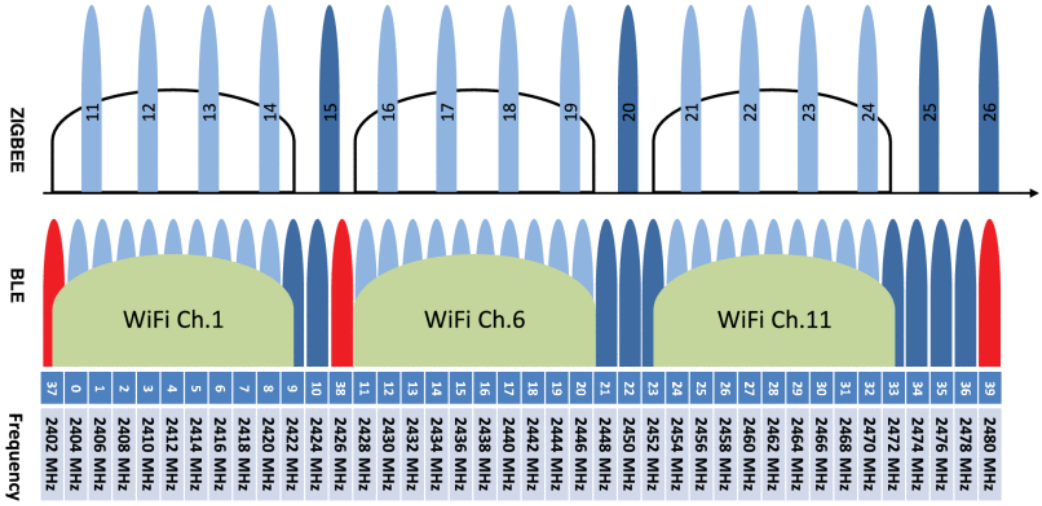

Figure 2-6. Coexistence des canaux BLE, Wi-Fi et Zigbee dans la bande 2.4 GHz, par courtoisie des auteurs de (La et al., 2018)

Pour assurer l'interopérabilité entre objets issus de différents fabricants, des « profils d'application » publics ont été introduits en fonction des cas d'usage : *Home Automation* ou *Industrial Plant Monitoring* par exemple. Le banc d'essai sur lequel j'ai mené mes expériences fonctionne avec le profil *Zigbee Light Link* (ZLL) dédié aux solutions

d'éclairage pour le grand public : suivi de la consommation, application de scénarios d'éclairage selon les présences et création d'ambiances lumineuses font partie des possibilités offertes par ce profil. ZLL fait partie de « Zigbee Pro », la version de Zigbee datant de 2007. Une version « Zigbee 3.0 » est apparue en 2015 ; rétro-compatible, son propos est d'offrir des associations d'objets plus sûres et d'améliorer l'interopérabilité entre objets Zigbee. Cela dit, j'ai été en mesure de vérifier que les dix objets d'éclairage Philips Hue achetés entre 2020 et 2022 pour bâtir le banc d'essai fonctionnaient toujours en 2022 avec le standard « Zigbee Pro » de 2007, un cas de figure fréquent dans l'IoT.

Comparé à d'autres profils comme *Home Automation*, ZLL est un profil plus simple. Par exemple, en ce qui concerne la sécurité, un réseau ZLL n'a pas de *Trust Center* ni de *Coordinator*, seulement un *Control Bridge* qui ne permet qu'une gestion basique des clés de sécurité. Cet élément joue aussi le rôle de passerelle Internet (*Internet Bridge*).

2.4.2 Pile Zigbee

La pile Zigbee est constituée de quatre couches comme indiqué Figure 2-7. Les deux couches basses sont la couche physique, que je désignerai comme telle ou par « couche PHY », et la couche liaison que je désignerai comme telle ou abusivement par « couche MAC » (*Media Access Control*)[6]. Les couches hautes sont les couches Réseau (*Network*) et Application. On peut qualifier cette pile de semi-ouverte car le standard des couches hautes est ouvert mais leurs implémentations sont propriétaires et fermées, dépendant de chaque fabricant.

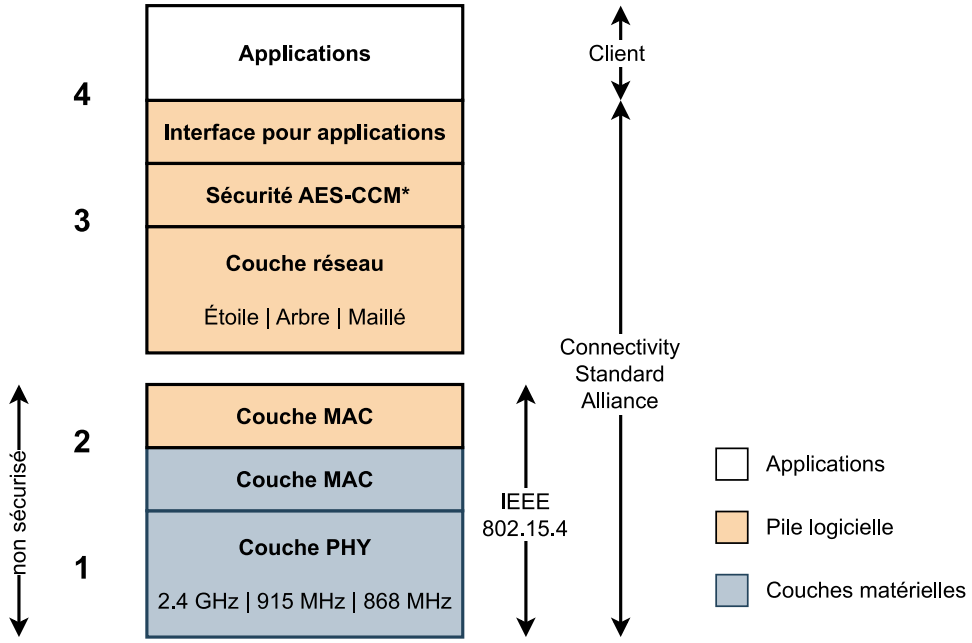

Figure 2-7. Pile Zigbee, d'après (Silabs, 2021)

[6] La couche liaison (couche 2 du modèle OSI) est en fait constituée de la sous-couche MAC à laquelle se superpose la sous-couche LLC 802.2 (*Logical Link Control*) (Hersent et al., 2014).

2.4.2.1 Couches basses

Zigbee Pro repose sur le standard IEEE 802.15.4-2003 pour les deux couches basses de la pile (PHY et MAC). Le format général de la trame IEEE 802.15.4 est donné Tableau 2-2, d'après une documentation Texas Instruments[7]. Les nombres indiqués correspondent à la taille de chaque champ, en octets. La longueur maximale de la trame (MPDU) est de 127 octets.

<table>
<tr><td rowspan="3">MAC Layer</td><td colspan="3" rowspan="2"></td><td>2</td><td>1</td><td>0 to 20</td><td>n</td><td>2</td></tr>
<tr><td>Frame Control Field (FCF)</td><td>Data Sequence Number</td><td>Address Information</td><td>Frame Payload</td><td>Frame Check Sequence (FCS)</td></tr>
<tr><td colspan="3">MAC Header (MHR)</td><td>MAC Payload</td><td>MAC Footer (MFR)</td></tr>
<tr><td rowspan="5">PHY Layer</td><td>4</td><td>1</td><td>1</td><td colspan="4">5 + (0 to 20) + n</td></tr>
<tr><td>Preamble Sequence</td><td>Start-of-Frame Delimiter (SFD)</td><td>Frame Length</td><td colspan="4">MAC Protocol Data Unit (MPDU)</td></tr>
<tr><td colspan="2">Synchronization Header (SHR)</td><td>PHY Header (PHR)</td><td colspan="4">PHY Service Data Unit (PSDU)</td></tr>
<tr><td colspan="7">5 + (0 to 20) + n</td></tr>
<tr><td colspan="7">PHY Protocol Data Unit (PPDU)</td></tr>
</table>

Tableau 2-2. Format général de la trame IEEE 802.15.4

2.4.2.2 Couches hautes

Description – Le standard Zigbee définit les deux couches hautes de la pile. S'appuyant sur une version modifiée du protocole de routage AODV (*Ad hoc On-demand Distance Vector*), la couche réseau permet des réseaux qui s'autoorganisent avec des topologies étoile, arbre et maillée (*mesh*). Cette dernière, illustrée Figure 2-8, permet l'extension de couverture et l'autoréparation en cas de route coupée. Les routeurs (*routers*) sont des prises et des ampoules connectées ou encore la passerelle Internet (*Internet bridge*), ils sont connectés au secteur. Dans ce cas, ils implémentent une pile Zigbee complète, dite *Full Function Device* (FFD). Au contraire, les télécommandes et différents capteurs (mouvement, température, luminosité) sont des nœuds terminaux (*end devices*) fonctionnant sur une batterie censée durer 10 ans. Ils implémentent des piles Zigbee partielles dites *Reduced Function Device* (RFD). Pour économiser l'énergie, chacun de ces objets « enfants » est la plupart du temps en mode sommeil. Quand il se réveille, il demande à son parent les données que celui-ci a reçues à son intention. La couche « interface pour applications » décrit un ensemble de messages normalisés par des clusters permettant de créer des applications distribuées sur plusieurs objets, créant ainsi des « liens » ou « liaisons » (*bindings*). Par exemple, une action sur une certaine touche d'une télécommande peut faire commuter deux prises intelligentes ou un passage devant un détecteur de mouvement peut faire allumer une ampoule avec une intensité dépendant de la luminosité courante.

[7] https://www.ti.com/lit/ug/swru191f/swru191f.pdf, page 217.

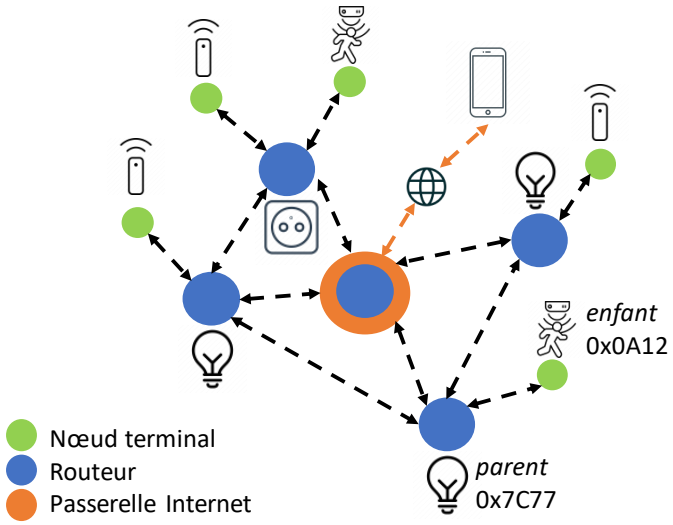

Figure 2-8. Exemple de réseau ZLL avec une topologie maillée

Sécurité – Les mesures de sécurité commencent seulement à partir de la couche réseau. Les différents messages Zigbee (de routage, de retour d'état, de commande d'allumage, etc.) sont envoyés par l'expéditeur chiffrés par une clé symétrique AES 128 bits appelé « clé de réseau » (*network key*). Cette clé est partagée par tous les objets du réseau, sa possession permet aussi de déchiffrer les messages ; sans elle, un récepteur ne voit qu'une trame IEEE 802.15.4 de type DATA dont les données sont chiffrées. Une clé différente par paire d'objets serait souhaitable, de même qu'une rotation régulière de la clé, mais ce n'est pas possible avec ZLL.

La Figure 2-9, d'après (Farahani, 2008), illustre le mécanisme décrit par le standard pour assurer la confidentialité et l'intégrité des données dans un échange. La confidentialité des données est assurée par le chiffrement de celles-ci côté émetteur grâce à un bloc AES-CCM*[8] paramétré par la clé de réseau et un *Nonce*. Outre le chiffrement des données, ce bloc calcule un code d'intégrité (*Message Integrity Code*, MIC), vérifié par le récepteur. Le *Nonce* fait figurer un compteur de trames sur 32 bits qui est incrémenté à chaque fois qu'une trame est transmise, ce qui constitue un dispositif anti-rejeu. Selon le niveau de sécurité recherché, il est possible de n'implémenter aucun des mécanismes précités, une partie ou la totalité. Des objets bon marché fonctionnant sur batterie, avec peu de capacité de calcul et de ressources mémoire, se dispensent généralement de ces protections (Stelte and Rodosek, 2013).

[8] AES-CCM* : *Advanced Encryption Standard-Counter with CBC MAC*.
CBC signifie lui-même *Cipher Block Chaining* et MAC *Message Authentication Code*.
L'astérisque dans AES-CCM* traduit le fait qu'on peut mettre en œuvre optionnellement : les mécanismes pour la confidentialité, les mécanismes pour l'intégrité ou les deux.

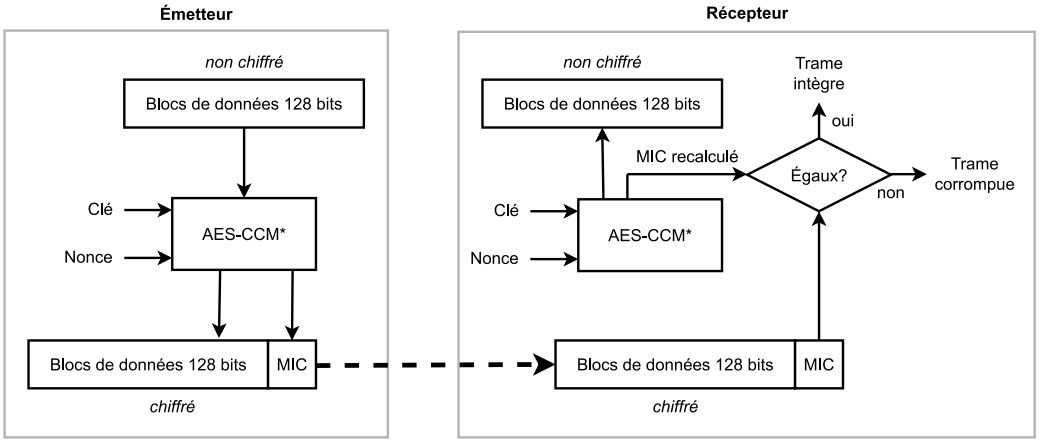

Figure 2-9. Confidentialité et intégrité des données dans un échange Zigbee

2.4.3 Vulnérabilités et attaques

En phase avec ce que j'ai expliqué dans la section 2.3.5, je ne considère que des attaques de communication au sein d'un réseau IoT. Je présente des vulnérabilités bien connues prenant place sur les couches basses et les couches hautes de la pile Zigbee et mettant en défaut les propriétés de sécurité définies en 2.1.2. Des exploitations de ces vulnérabilités par des attaques sont complètement détaillées dans le chapitre portant sur la réalisation d'un jeu de données Zigbee.

2.4.3.1 Vulnérabilités des couches basses

Écoute passive – Un attaquant équipé d'un démodulateur écoutant sur le bon canal d'échange peut acquérir toutes les trames IEEE 802.15.4. Certes, les données transportées par ces trames (qui sont, pour faire simple, les Unités de Données Protocolaire (PDU) de la couche réseau) sont chiffrées mais toutes les données d'en-tête et de queue de ces trames sont en clair, ce qui permet d'obtenir facilement des informations sensibles : nombre d'objets, identifiants sources et destinations, longueur des trames émises, temps inter-trame pour chaque objet, etc. L'exploitation de ces données permet de caractériser l'état du réseau et par-là d'inférer des comportements de l'utilisateur comme sa présence effective dans le logement, la mise en route de systèmes, etc. On peut parler d'atteinte à la vie privée. On ne peut pas faire grand-chose contre une écoute passive quand le protocole n'a pas été pensé pour s'y opposer. Le protocole Bluetooth peut se protéger contre l'écoute passive depuis sa version 4.2.

Absence d'authentification – Aucun mécanisme d'authentification n'est prévu pour empêcher un attaquant d'usurper l'identité d'un nœud légitime pour effectuer des requêtes de données vers un nœud victime en forgeant des trames avec l'adresse du nœud légitime comme adresse source et l'adresse du nœud victime comme adresse de destination (attaque de type mascarade, *masquerade attack* en anglais). L'attaquant peut inonder la victime de requêtes de données (*data request flood*) en apparence légitimes, causant un épuisement des ressources de celle-ci et du réseau, pouvant aller jusqu'à un déni de service.

Association confiée à la couche MAC – Un attaquant peut se faire passer pour un objet voulant s'associer au réseau en envoyant une commande *Association Request* au niveau de la couche MAC. Il obtiendra un identifiant 16 bits après avoir donné son adresse MAC 64 bits. Le nombre d'identifiant n'est pas infini, si l'attaquant effectue ce type de requête en

boucle en arborant à chaque fois une nouvelle adresse MAC (attaque de type sybil, *sybil attack* en anglais), il finira par épuiser l'ensemble d'identifiants disponibles.

2.4.3.2 Vulnérabilité des couches hautes

Clef de provisionnement connue de tous – Un objet s'associe au réseau pendant ce qu'on appelle la phase de provisionnement. Je me concentre ici sur la toujours disponible et répandue procédure d'association Zigbee permise par IEEE 802.15.4. Pour s'associer, un objet qui ne l'est pas encore émet à sa mise sous tension des trames IEEE 802.15.4 *Beacon Request* sur tous les canaux. Le premier routeur dans le voisinage qui répond avec une trame *Beacon* va gérer le processus d'association. À cette occasion, l'objet, qui n'avait pour l'instant qu'une adresse MAC 64 bits, va recevoir un identifiant 16 bits (moins encombrant pour la suite des échanges) ainsi que la clé de réseau, mentionnée plus haut, chiffrée par une « clé de provisionnement » (*light link commissioning key*), unique et préinstallée sur tous les objets ZLL du monde. Il n'est même pas besoin d'une attaque matérielle sur l'objet pour acquérir cette clé puisqu'elle est accessible par une simple recherche sur le net. Par la suite, une écoute passive d'un processus d'association, grâce à Wireshark renseigné avec la clé de provisionnement, donne immédiatement la clé de réseau. C'est toujours d'actualité en 2023 par respect de la rétrocompatibilité. C'est une vulnérabilité majeure de sécurité car aussitôt qu'un attaquant possède cette clé, il déchiffre en clair tous les messages des couches hautes et peut aussi compromettre l'ensemble du réseau (Zillner, 2016). Pour forcer un utilisateur à réaliser une association d'objet, on peut par exemple inonder l'objet de messages (cf. 2.4.3.1) ou brouiller les communications du réseau. Un provisionnement dit « Touchlink » plus sûr est apparu mais un bug décrit dans (Ronen et al., 2017) a été découvert par ses auteurs. Il a été corrigé depuis.

2.4.4 Synthèse

Étant donné le contexte de faible sécurité exposé lors de cette présentation du profil Zigbee Light Link, il apparait important d'authentifier les objets avec des éléments plus sûr qu'un identifiant facile à forger dans une trame ou la possession effective d'une clé de réseau qu'il est facile de se procurer. Les attaques par usurpation d'identité pourraient ainsi être détectées, ce qui n'est pas le cas actuellement.

2.5 IDS : une taxonomie

2.5.1 Pare-feux et IDS

Nous avons vu que le modèle IoT présente des caractéristiques le différentiant fondamentalement de l'informatique traditionnelle. Au final, les vulnérabilités sont aussi différentes et il y a lieu de lui réserver des solutions de sécurité dédiées.

Les systèmes de l'informatique traditionnelle (basés sur TCP/IP) sont souvent équipés de pare-feux pour contribuer à assurer leur sécurité. Ceux-ci peuvent être « réseau » pour gérer l'exposition des différentes parties d'un réseau ou « hôte » pour gérer l'exposition d'une interface réseau. Ils sont basés sur la définition de règles simples utilisant les notions d'adresses IP, ports et protocoles, afin d'autoriser ou non les communications. Dans chaque cas, les parties qui restent exposées peuvent être protégées respectivement par un IDS réseau (par exemple Snort / Suricata pour de la détection de motifs réseau malicieux) ou par un IDS hôte (par exemple Fail2ban pour la détection et mitigation d'attaques SSH par force brute). Les IDS permettent l'utilisation de modèles fins, complexes et puissants, notamment par l'usage de l'apprentissage automatique.

Comme nous l'avons vu en section 2.1.1.2, les objets IoT à pile Wi-Fi peuvent bénéficier du pare-feu réseau associé au réseau dans lequel ils s'insèrent, ce qui peut être intéressant pour bloquer l'accès à certains services d'un objet (par l'interface de gestion de la box

internet du domicile par exemple). Par contre, contrairement à de véritables postes informatiques, il est rarement possible de leur adjoindre un logiciel pare-feu hôte pour des raisons de ressources, d'hétérogénéité et d'ouverture.

Quand les réseaux sont décentralisés comme c'est le cas des réseaux *ad hoc* du standard Zigbee, un pare-feu réseau unique surveillant l'ensemble des nœuds n'est d'ailleurs même pas possible puisque des communications peuvent lui échapper. En outre, un pare-feu hôte pour chaque objet semble aussi peu réaliste qu'avec des objets Wi-Fi. Enfin, la transposition d'un pare-feu TCP/IP pour un protocole comme Zigbee reste pleinement à définir. Par la suite, je vais me consacrer uniquement aux IDS pour l'IoT, les pare-feux réseaux tels que nous les connaissons n'étant envisageables que pour les seuls objets Wi-Fi.

2.5.2 Définition et constitution d'un IDS

Un système de détection d'intrusion (IDS) surveille dynamiquement les actions qui ont lieu dans un environnement et décide si celles-ci sont symptomatiques d'une attaque ou constituent un usage légitime de cet environnement (Debar et al., 1999).

Dans cette définition générale, un « environnement » peut être un réseau IoT ou bien particulièrement un hôte IoT. Typiquement, un IDS se décompose en trois fonctions, à l'image de ce qu'illustre la Figure 2-10 :

- la collecte d'informations en lien avec la détection d'intrusion, à l'aide d'une ou plusieurs sondes ;
- l'analyse de ces informations pour décider s'il y a intrusion ou pas ;
- la génération d'alarme en cas d'intrusion estimée, éventuellement associée à une journalisation.

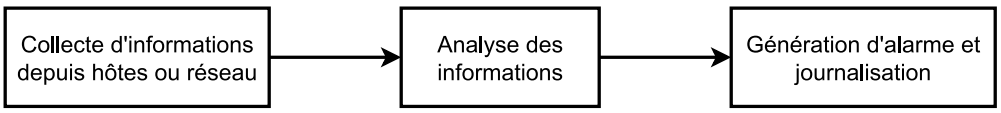

Figure 2-10. Décomposition des fonctions d'un IDS

Une fois qu'une attaque est détectée, l'IDS peut chercher à en réduire les effets voire la bloquer grâce à des actions de mitigation. Dans ce cas, on parle d'IDS/IPS (*Intrusion Detection System / Intrusion Prevention System*). La plus simple de ces contremesures consiste sans doute à mettre en quarantaine l'objet ou la partie de réseau infectée par l'intrusion.

2.5.3 Métriques de performance d'un IDS

Un IDS est, dans sa réalisation la plus simple, un classificateur binaire qui effectue un travail de prédiction afin de ranger les situations dans deux classes : « intrusion » et « absence d'intrusion ».

Pour mesurer la performance d'un système de détection, on quantifie d'abord ses vrais et faux positifs ainsi que ses vrais et faux négatifs. Les considérations qui suivent sont issues de (Mitchell and Chen, 2014) et de (Géron, 2019) :

- Quand un IDS prédit une intrusion et qu'effectivement il y en a une, c'est un vrai positif (*true positive*).
- Quand un IDS prédit une absence d'intrusion et qu'effectivement il n'y en a pas, c'est un vrai négatif (*true negative*).
- Quand un IDS prédit une intrusion alors qu'il n'y en a pas c'est un faux positif (*false positive*) ; l'excès de faux positifs perturbe l'utilisateur, lui fait perdre confiance en son IDS au point parfois de renoncer à s'en servir.
- Quand un IDS prédit une absence d'intrusion alors qu'il y en a une, c'est un faux négatif (*false negative*). Ce cas est le plus dangereux puisque l'utilisateur se croit en sécurité alors qu'une attaque a lieu.

Notons *TP*, *FP*, *TN* et *FN* le nombre de vrais positifs, faux positifs, vrais négatifs et faux négatifs obtenus lors d'une campagne de test destinée à mesurer la performance d'un IDS. Un IDS fonctionnant parfaitement a des *TP* et *TN* non nuls et des *FP* et *FN* nuls.

Outre l'obtention d'une matrice de confusion, ces quatre scalaires permettent de générer des métriques assez synthétiques.

La première d'entre elles est la justesse (*accuracy*) qui mesure le ratio de prédictions correctes sur la totalité des cas :

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2-1}$$

Mais dans certains contextes, cette métrique peut être inadaptée[9]. On préfère donc définir également le taux de vrai positif (*True Positive Rate*, *TPR*) appelé aussi rappel (*recall*), ainsi que la précision (*precision*). Les formules sont les suivantes :

$$recall = TPR = \frac{TP}{TP + FN} \tag{2-2}$$

$$precision = \frac{TP}{TP + FP} \tag{2-3}$$

recall donne le ratio du nombre de prédictions d'intrusion correctes sur le nombre total d'intrusions. Il caractérise en quelque sorte l'efficacité de l'IDS à prédire toutes les intrusions. On cherchera à le rendre le plus proche de 1 si on ne veut rater aucune intrusion, quitte à avoir des faux positifs.

precision donne le ratio du nombre de prédictions d'intrusion correctes sur le nombre total de prédictions positives. Il caractérise en quelque sorte la pertinence des prédictions positives. On cherchera à le rendre le plus proche de 1 si on veut qu'une prédiction d'intrusion corresponde effectivement à une intrusion, quitte à avoir des faux négatifs.

recall et *precision* évoluent en sens contraire. Quand l'un augmente, l'autre diminue. En général, soit on choisit un compromis, soit la situation impose de maximiser l'un aux dépens de l'autre.

[9] C'est le cas quand les probabilités d'occurrence des évènements des différentes classes sont très différentes.

Il est parfois intéressant de donner le taux de vrais négatifs (*True Negative Rate*, *TNR*). Celui-ci correspond au ratio du nombre de prédictions d'absence d'intrusion correctes sur le nombre total d'absences d'intrusion :

$$TNR = \frac{TN}{TN + FP} \tag{2-4}$$

Enfin, le taux de faux positifs (*False Positive Rate*, *FPR*) traduit le nombre de faux positifs sur le nombre total d'absences d'intrusion. C'est le complément à 1 du *TNR* :

$$FPR = \frac{FP}{TN + FP} \tag{2-5}$$

2.5.4 Taxonomie des IDS

La production scientifique met à disposition de nombreuses revues de littérature concernant la détection d'intrusion dans l'IoT. À l'image des travaux exposés dans (Mitchell and Chen, 2014), (Zarpelão et al., 2017) et (Benkhelifa et al., 2018), celles-ci proposent leur propre taxonomie des IDS. Les critères de classement proposés sont variables mais néanmoins, deux sont souvent présentés en premier. Il s'agit de la stratégie de placement des IDS et des méthodes de détection utilisées par ceux-ci. Dans une moindre mesure, on trouve le type de menaces traitées comme troisième critère. Le Tableau 2-3 répertorie selon les stratégies de placement et les méthodes de détection quelques travaux représentatifs parmi ceux que j'ai étudiés. Ceux-ci sont repris dans le texte de cette section pour illustrer mon propos.

		Détection		
		Signature	**Hybride**	**Anomalie**
Placement	**Distribué**	(Oh et al., 2014)		(T.-H. Lee et al., 2014)
	Hybride		(Raza et al., 2013)	
	Centralisé	(Kasinathan et al., 2013)		(Siby et al., 2017) (Roux et al., 2018)

Tableau 2-3. Placement de quelques travaux dans la taxonomie des IDS

2.5.4.1 Stratégies de placement des IDS

Ce critère rend compte de l'architecture de l'IDS global retenu. On peut distinguer trois stratégies de placement dans les configurations les plus courantes.

Placement distribué – Dans ce cas de figure, chaque objet intègre son propre IDS, appelé alors HIDS (*Host-based IDS*). Vu sa position, l'HIDS peut collecter de l'information riche et intime concernant l'objet. Les caractéristiques de son trafic réseau (nombre de connexions répétées, bande passante consommée, etc.) sont souvent exploitées ; un accès à de l'information déchiffrée peut même être envisagé si le HIDS est intégré à la conception de l'objet. Les changements au niveau du système d'exploitation (appels système,

modification du système de fichiers, etc.) ou des applications peuvent être obtenus via des journaux. La surveillance de canaux auxiliaires (consommation électrique ou température) est aussi envisageable comme cela est fait dans (T.-H. Lee et al., 2014). Les HIDS doivent cela dit être conçus pour les différents systèmes d'exploitation des hôtes et ils en consomment les ressources déjà faibles. Le cas échéant, la non-ouverture de certaines parties du code rend impossible la synthèse de ce type d'outil de sécurité. Au final, un HIDS fonctionnel pour objet contraint, pouvant de surcroit être mis à jour à distance, reste un but difficilement atteignable. Un exemple de placement distribué est (T.-H. Lee et al., 2014) dont le but est de détecter les attaques DoS dans un contexte 6LoWPAN. Plutôt qu'analyser des trames, une détection par observation d'un canal auxiliaire a été mise en place. En effet, chaque nœud compare sa consommation électrique par rapport à un modèle normal de fonctionnement. En cas d'anomalie de consommation, le nœud est considéré victime d'une attaque DoS. Cette solution a été conçue pour ne solliciter que peu de ressources sur chaque nœud.

Placement centralisé – Dans ce cas, la totalité de l'IDS est logé dans un composant « central ». Celui-ci a vocation à surveiller tout ou partie du réseau et on le nomme alors NIDS (*Network-based IDS*). Le NIDS peut s'intéresser à des aspects classiques des trames échangées comme leurs adresses source et destination, leur longueur, le temps les séparant, etc. Il peut aussi s'intéresser à la puissance de celles-ci via le RSSI (*Received Signal Strength Indicator*) mesurée sur la sonde de l'IDS ou encore à des caractéristiques physiques très fines comme la longueur du préambule d'une trame pour identifier les objets (Helluy-Lafont et al., 2020). Le NIDS ne consomme généralement pas de ressource sur les objets puisqu'il n'est pas installé sur eux. Il peut être dimensionné selon les tâches qui lui incombent et des mises à jour de son logiciel sont envisageables, surtout s'il a un accès à Internet. Le NIDS doit être placé de façon à ce qu'il ait accès au plus grand nombre d'échanges entre les nœuds. Dans des réseaux centralisés de type Wi-Fi, le point d'accès parait tout indiqué puisque toutes les trames passent par lui ; le NIDS est alors un nœud effectif du réseau. Dans le cas de réseaux *ad hoc* comme c'est le cas des réseaux Zigbee, il n'est pas de placement central « naturel ». Dans ce cas, on peut intégrer le NIDS dans le nœud effectif du réseau par où passe le plus de trames (typiquement, le pont Internet) mais cela doit s'envisager dès la conception. Néanmoins, certains échanges ne seront pas vus étant donnée la topologie. Il est aussi possible d'envisager un NIDS « furtif », ne faisant pas réellement partie du réseau et fonctionnant en pure écoute passive, même si ce n'est pas toujours possible avec certains protocoles. Ce principe est exploité dans les solutions IoTScanner (Siby et al., 2017) et RadIoT (Roux et al., 2018). La première utilise plusieurs sondes dédiées à différentes technologies (BLE, Wi-Fi, Zigbee) afin d'obtenir des données démodulées, donc facilement traitables ; la seconde utilise une radio logicielle (*Software-Defined Radio*, SDR) en mode balayage pour mesurer la puissance globale sur des bandes de fréquences. Enfin, si le réseau est particulièrement étendu ou présente beaucoup d'obstacles, il est probable qu'une seule sonde ne suffise pas à couvrir tout l'espace surveillé (ou à donner assez d'information). Envisager plusieurs sondes distribuées communiquant avec l'IDS central est sans doute nécessaire mais les considérations d'architecture et de synchronisation des informations passent à une autre dimension de complexité.

Placement hybride – Ce placement, illustré par la solution SVELTE exposée dans (Raza et al., 2013), tente d'exploiter les points forts des deux stratégies précédentes afin d'obtenir de meilleures métriques de détection, Dans ce cas, les HIDS interagissent avec le NIDS central. Cela se fait au prix d'une réalisation beaucoup plus exigeante, pas forcément réaliste dans les contextes qui m'intéressent.

Nous pouvons remarquer que j'ai associé HIDS à placement distribué et NIDS à placement centralisé. Ce sont des cas de figures fréquents mais d'autres combinaisons existent. Par

exemple, on peut imaginer plusieurs NIDS distribués si le réseau à surveiller est très grand. Également, un hôte hyperviseur de plusieurs machines virtuelles peut surveiller celles-ci de façon centralisée.

2.5.4.2 Méthodes de détection des IDS

Cette section présente les deux méthodes de détection communément utilisées dans les IDS.

Détection basée sur les signatures – La détection à base de signature (*signature-based detection* ou *misuse detection*) utilise une base de données des attaques connues. Chacune de ces attaques possède une signature que l'on retrouve dans la charge utile des messages la véhiculant. Cette méthode fonctionne bien pour détecter des attaques connues et présente donc un faible nombre de faux positifs. Par contre, la détection des nouvelles attaques comme les attaques *zero day*, nombreuses dans l'IoT, est logiquement mauvaise. Ainsi, sous peine d'obtenir beaucoup de faux négatifs, cette méthode réclame donc un travail régulier d'actualisation de la base des signatures. Cette contrainte semble difficilement conciliable avec la forte hétérogénéité des protocoles de l'IoT. Des réalisations existent cela dit. Par exemple, le NIDS présenté dans (Kasinathan et al., 2013) exploite la base de signatures actualisée de l'IDS Suricata dans un contexte 6LoWPAN. Les auteurs de (Oh et al., 2014) ont implémenté un algorithme allégé de filtrage par motif (*pattern matching*) dans des hôtes mais ceux-ci sont des nano-ordinateurs relativement puissants. Généralement, la méthode de détection par signature, à la base des antivirus dans l'IT, n'est pas implémentable dans des hôtes contraints, eu égard 1) à l'empreinte mémoire de la base de signatures, 2) aux ressources de traitement requises par l'exécution des algorithmes de filtrage par motif généralement utilisés et 3) au déficit d'ouverture de certains objets. Il n'existe toujours pas d'antivirus pour ampoule connectée (Helluy-Lafont, 2021).

Détection basée sur les anomalies – La détection basée sur les anomalies (*anomaly-based detection* ou *behavioural detection*) compare le comportement d'un système à un moment donné avec celui prédit par un modèle de comportement normal de ce système. Quand il y a un écart entre les deux supérieur à un certain seuil, l'IDS signale une anomalie, considérée comme un symptôme d'intrusion ou d'attaque. Le comportement plutôt régulier des réseaux IoT se prête bien à ce type de détection. Le modèle doit être alimenté par des « traits », « caractéristiques », « propriétés », « indicateurs » ou encore « attributs » (en anglais : *features*) caractérisant bien le système à modéliser et le type d'attaque que l'on veut détecter. Par exemple, le temps inter-trames que l'on peut fabriquer à partir des estampilles (*timestamps*) des trames est un bon révélateur des attaques DoS. Par la suite, j'utiliserai le terme « attribut ». L'approche peut être statique ou intégrer des séquences chronologiques. Par son principe, la détection d'anomalies repère bien les nouvelles attaques (y compris celle de type *zero day*) et jouit donc d'un faible nombre de faux négatifs. Mais définir un modèle complet de comportement normal n'est pas simple et la méthode peut souffrir d'un grand nombre de faux positifs. La détection d'anomalies, si elle doit être distribuée dans chaque hôte contraint, ne peut qu'utiliser des heuristiques basiques comme dans (T.-H. Lee et al., 2014) où chaque nœud compare sa consommation par rapport à la normale, pour être retiré de la table de routage si l'écart entre les deux est trop grand, dans le but de combattre les attaques par déni de service. Cela dit, dans l'IT comme l'IoT, la tendance massive pour la détection d'anomalies est à l'apprentissage automatique, ce qui remplace l'établissement coûteux de modèles établis par un expert, même si cette technique (appelée en anglais *specification-based detection*) donne moins de faux positifs et ne nécessite pas d'entraînement (Zarpelão et al., 2017). Les algorithmes de l'apprentissage automatique pour la détection d'intrusions peuvent consommer une quantité importante de ressources et ne sont pas gérables par des hôtes contraints ; les algorithmes appropriés sont dans ce cas à intégrer dans un NIDS correctement dimensionné. Ainsi, celui de RadIoT (Roux et al., 2018) prédit une attaque à l'aide d'un réseau de neurones à auto-encodeur.

Celui-ci a été entrainé à reconstruire en situation sans attaque la puissance électromagnétique captée par la sonde du NIDS par spectrogrammes de 100 MHz à partir de quelques données statistiques de chaque spectrogramme. Ainsi, en situation de détection, un écart supérieur à un certain seuil entre la reconstruction et la réalité est caractéristique d'une attaque.

Détection hybride – Au prix d'une grande complexité, on peut faire intervenir les deux types de détection. Les auteurs de la solution SVELTE exposée dans (Raza et al., 2013) annoncent des détections d'attaques de routage satisfaisantes en ayant un bon compromis coût du stockage de la base de données / coût de traitement lié à détection d'anomalies.

2.5.5 Destinations des IDS à détection d'anomalie

Je donne ci-après les principaux usages des IDS à détection d'anomalie, ceux à détection de signatures restant plus limités.

2.5.5.1 IDS détectant des attaques sans besoin d'identifier les objets

Ceux-ci exploitent des agrégations comme la puissance radioélectrique globale des réseaux IoT présents dans une maison. On peut citer à titre d'illustration le travail original décrit dans (Roux et al., 2018), dont nous avons déjà parlé. N'étant pas sensible aux protocoles utilisés, cette solution couvre naturellement des environnements avec différentes piles de protocoles et peut en intégrer de nouvelles. Au final, cette solution élégante détecte bien les attaques DoS et les points d'accès Wi-Fi pirates mais les attaques concernant le réseau Zigbee sont très mal repérées eu égard à la faible puissance d'émission de cette technologie comparée au Wi-Fi. En outre, des attaques devant respecter la sémantique d'un protocole sont probablement mal détectées. Enfin, son approche agrégative fait sa simplicité mais aussi un de ses défauts : Quand une attaque a lieu, si celle-ci est signalée, on ne peut savoir que la fréquence à laquelle elle se produit mais il n'est pas possible de savoir quel objet a été corrompu.

2.5.5.2 IDS détectant des attaques

Ce type de travaux est très répandu dans la littérature. Dans ceux-ci, l'IDS extrait en continu des informations jugées pertinentes pour la détection considérée. En général les informations sont regroupées par objet et par fenêtre temporelle sur laquelle des statistiques (moyenne, écart type, etc.) sont effectuées. Le résultat de cette extraction-traitement permet d'obtenir les attributs (*features*) à destination du modèle de comportement (typiquement un classificateur) de chaque objet afin d'inférer s'il y a une attaque ou pas. Pour mener ce travail, les objets sont discriminés par une identité dite « faible » qualifiée ainsi car facilement forgeable, telle une adresse MAC ou IP ou un identifiant source sur 16 bits dans une trame Zigbee. Ainsi, les travaux décrits dans (Doshi et al., 2018) ont pour ambition de détecter dans un réseau interne quels objets sont en train de pratiquer un DoS ou un DDoS ; les auteurs ont retenu des attributs (phase de *feature engineering*) à l'aide de fonctions cumulatives de distribution (*Cumulative Distribution Functions*, CDF). Ainsi, les attributs numériques issus des données de couche MAC comme la taille des trames, le temps inter-trames et la bande passante semblent pertinents pour l'objectif à atteindre, au même titre que l'attribut catégoriel « plus haut protocole présent dans la trame ». Les auteurs ont utilisé l'apprentissage supervisé, ce qui suppose des données d'entraînement, et ils comparent les résultats de divers algorithmes détectant les attaques. Ceux-ci sont pour la plupart excellents. La pile considérée est de type Wi-Fi.

2.5.5.3 IDS identifiant le type des objets

On trouve beaucoup d'IDS de ce type dans la littérature car il est considéré à juste titre que la sécurité d'un réseau passe par une bonne surveillance (monitoring) de ses objets et de

leur rôle. Un type d'objet correspond par exemple au triplet <*marque, référence, version de micrologiciel*>. En général, les caractéristiques disponibles sur les couches MAC et supérieures des trames sont utilisées pour générer les attributs requis par le modèle inférant le type de l'objet.La génération d'une empreinte basée sur des caractéristiques réseau des objets est souvent mise en œuvre dans ce type d'IDS. La solution ProfillIoT (Meidan et al., 2017) identifie les types d'objets en se basant sur le trafic TCP. IoTSense (Bezawada et al., 2018) fait de même en utilisant la taille de fenêtre TCP ainsi que longueur et entropie de la charge utile. IoTSentinel (Miettinen et al., 2017) propose d'identifier le type des objets apparaissant dans le réseau par des classificateurs travaillant sur le trafic réseau initial de ces objets, afin de voir si aucun ne fait l'objet d'une des vulnérabilités stockées dans une base externe (la maintenance de celle-ci pose question vu le rythme de sortie des objets IoT). Ces solutions sont intéressantes mais elles travaillent toutes trois uniquement avec la pile Wi-Fi. Ensuite, la connaissance préalable de certaines vulnérabilités est parfois nécessaire. Enfin, elles utilisent l'apprentissage supervisé, ce qui réclame des données d'entraînement de qualité, donc coûteuses. Au contraire, la solution IoTHound (Anantharaman et al., 2020) regroupe les objets par type grâce à un algorithme non-supervisé de partitionnement (*unsupervised clustering*) HDBSCAN, en utilisant des attributs comme le temps inter-trames et l'entropie des données. Par la suite, un écart d'un objet de sa partition de type témoigne d'une anomalie et donc probablement d'une intrusion. En outre, grâce à des composants sur étagère bon marché, elle gère trois piles IoT, ce qui est un pas significatif dans la gestion de l'hétérogénéité des technologies. Il faut cela dit être conscient que toutes les techniques présentées dans ce paragraphe suivent les objets par des identités « faibles », comme les IDS du paragraphe précédent. Peut-on leur faire toujours confiance ? Non, et c'est une grave limitation car les attaques par usurpation ne sont pas détectables et elles amènent de fausses informations à l'IDS. Ce doute est présent de l'extraction des attributs par tri sur l'identifiant jusqu'au relevé de l'identifiant de l'objet présentant une anomalie. Par ailleurs, beaucoup de travaux dans l'esprit de ce paragraphe sont illustrés par des bancs d'essai avec plusieurs types d'objets mais une seule instance pour chacun de ces types, ce qui n'est pas réaliste. En effet, il est courant dans une maison d'avoir dix ampoules du même type. Ainsi, avec ce type de configuration biaisée, il est facile, même involontairement, d'annoncer une identification d'instance alors que ce n'est qu'une identification de type.

2.5.5.4 IDS identifiant les instances d'objets

Par ses caractéristiques, l'IoT a amené de nouveaux défis dans l'identification – « quel objet est-ce ? » – et leur authentification – « est-ce que l'objet est celui qu'il prétend être ? » (Bezawada et al., 2018). Ces travaux sont rares dans la littérature comme le confirme la revue (Jmila et al., 2022). Identifier ou authentifier les objets individuellement est nécessaire pour permettre à un administrateur d'isoler un objet infecté par un logiciel malveillant. De même, il est important de détecter les attaques par usurpation d'identité. Ainsi, sur les couches où l'authentification cryptographique en relation est absente ou peu viable, une identité « faible » (*soft identity*) ne peut suffire et une authentification de complément permettant une identité « forte » (*strong identity*) est nécessaire. Par ailleurs, dans la pile TCP/IP, les adresses MAC font parfois l'objet d'une anonymisation en étant régulièrement redistribuées pour le respect de la vie privée, ce qui rend délicat le suivi d'un objet par cet identifiant. L'élaboration d'une empreinte physique par objet (*device fingerprinting*), si possible difficilement imitable par un attaquant, est un thème porteur dans la problématique de l'identification. M'intéressant à cette dernière dans le cadre vulnérable de la maison connectée, la section suivante est consacrée aux empreintes d'objets. La technologie plus jeune des PUF (*Physical Unclonable Functions*), consistant à identifier un objet de manière unique sur des propriétés mécaniques via un modèle défi-

réponse, semble prometteuse mais reste encore trop expérimentale et onéreuse pour l'IoT que je considère.

2.6 Empreintes d'objets

La technique d'empreinte consiste à récolter des informations sur un objet pour en générer une signature propre, afin de l'utiliser pour identifier celui-ci. Dans la revue de littérature (Xu et al., 2016) consacrée à cette technique, les auteurs recensent d'abord quels attributs peuvent être retenus puis les algorithmes pouvant les exploiter. Le Tableau 2-4 répertorie selon ces deux critères cinq travaux variés représentatifs du domaine, commentés au fil du texte de cette section. La pile utilisée est aussi mentionnée pour chaque travail.

			Attributs		
			Radiométriques	**Dépendants de l'emplacement**	
				RSSI	**PSD**
Algorithmes	**Liste blanche**	**Classification**	(Brik et al., 2008), Wi-Fi		
		Similarité			(Galtier et al., 2020), BLE & Zigbee
	Non-supervisé	**Trivial**		(Faria and Cheriton, 2006), Wi-Fi	
		Partitionnement (*clustering*)	(Nguyen et al., 2011), Zigbee		
		Entraînement sur une seule classe		(Madani and Vlajic, 2021), IEEE 802.15.4	

Tableau 2-4. Quelques travaux étudiés concernant l'authentification et la détection d'usurpation par empreinte

2.6.1 Attributs pour les empreintes d'objets

De façon synthétique, les attributs extraits depuis des informations contenues dans les couches MAC et supérieures (e.g., longueur de trame, bande passante, temps inter-trames, entropie) permettent d'inférer des types d'objets. Par contre, pour identifier des instances

d'objets, une extraction d'attributs issus d'informations de la couche physique est nécessaire. Me préoccupant de cette problématique, je donne ci-après un classement des attributs de couche PHY. Ils sont de deux types : radiométriques et dépendants de l'emplacement.

2.6.1.1 Attributs radiométriques

Ces attributs, nombreux, non dépendants de l'emplacement, reposent sur des imperfections de la couche physique introduites pendant la fabrication de l'émetteur-récepteur. On distingue les attributs radiométriques basés sur la forme d'onde et ceux basés sur la modulation du signal. Les premiers permettent de caractériser de façon unique une instance d'objet parmi plusieurs d'un même type, en capturant par exemple le signal transitoire d'allumage, très distinctif. Cela nécessite cependant une fréquence d'échantillonnage élevée, de l'ordre du GHz, et la sensibilité au bruit est forte. Avec les attributs basés sur la modulation, les signaux sont représentés sous forme d'échantillons démodulés (I, Q). À l'œuvre dans une SDR, cette technique utilisant le principe du récepteur superhétérodyne permet un échantillonnage à seulement quelques dizaines de MHz. Les attributs liés à la modulation sont mieux structurés et plus simples à manipuler mais sont cela dit moins précis. Ainsi ils peuvent avoir du mal à différencier deux instances d'un même type d'objet (Helluy-Lafont, 2021). À titre d'illustrations, le travail décrit dans (Nguyen et al., 2011) exploite une différence de fréquence porteuse et une différence de décalage de phase par rapport à des références. Les travaux de la solution PARADIS (Brik et al., 2008) exploitent les versions démodulées des signaux pour en reconstruire une version modulée idéale afin de la comparer aux signaux réels.

Les points forts des attributs radiométriques sont :

- la non dépendance à l'emplacement de l'objet, celui-ci peut raisonnablement bouger (tant qu'il reste vu par l'élément d'acquisition) ;
- la non dépendance aux changements d'environnements, dus par exemple à la présence d'habitants qui se déplacent ou à des éléments qui changent de configuration (meubles, portes, etc.).

Par contre, la phase d'extraction des attributs pour constituer l'empreinte d'un objet présente des inconvénients :

- un travail exigeant d'ingénierie en traitement du signal, basé sur du matériel spécialisé, par exemple une radio logicielle (SDR) dans le cas de l'utilisation d'attributs issus du domaine de la modulation. Les données manipulées, au format (I, Q), présentent un volume conséquent de plusieurs Mio par seconde de capture. Globalement le traitement est coûteux et pas toujours automatisable, il peut prendre un temps non compatible avec des exigences de détection temps réel. Les SDR sont abordées plus en détail dans ma première contribution portant sur la conception d'un IDS réaliste ;
- une dépendance à la technique de modulation utilisée, ce qui ne permet pas de généraliser l'extraction en cas d'environnement IoT hétérogène.

2.6.1.2 Attributs dépendants de l'emplacement : RSSI et PSD

Le RSSI est le principal attribut de couche physique de type « dépendant de l'emplacement » disponible. Il a été retenu comme attribut dans (Faria and Cheriton, 2006) et (Madani and Vlajic, 2021).

Le RSSI correspond à une mesure en dBm de la puissance moyenne[10] d'une trame issue d'un émetteur lorsqu'elle arrive sur un récepteur, par exemple une sonde. Dans des conditions théoriques d'espace libre, cette puissance est proportionnelle à la puissance de l'émetteur P_e exprimée en mW et inversement proportionnelle au carré de la distance émetteur-récepteur d. Dans la formule (2-6) donnant l'expression du RSSI, le facteur de proportionnalité k intègre les gains des antennes d'émission et de réception, il dépend aussi de la fréquence de fonctionnement :

$$RSSI = 10\log\left(k\frac{P_e/1\text{mW}}{d^2}\right) \quad (2\text{-}6)$$

Si émetteur et sonde réceptrice restent fixes et que l'émetteur ne change pas sa puissance d'émission, la sonde mesure toujours la même valeur de RSSI pour cet émetteur ; ce peut être le point de départ d'une réflexion sur une authentification basée sur cet attribut physique.

Les points forts du RSSI sont :

- son accès immédiat : En général, dans les récepteurs dédiés à une technologie, le RSSI est disponible sous la forme d'un champ dans l'en-tête ou le pied des trames. Aucun travail ni matériel complexes et coûteux d'extraction n'est nécessaire ;
- sa disponibilité sur la plupart des protocoles : Le RSSI est présent dans tous les protocoles nécessitant une évaluation de canal libre (*Clear Channel Assessment*), c'est-à-dire ceux utilisant CSMA/CA (*Carrier-Sense Multiple Access/Collision Avoidance*), ce qui est le cas des protocoles sans-fil de l'IEEE. Le RSSI est disponible également avec BLE ;
- sa simplicité : Le RSSI se présente sous la forme d'un simple scalaire, laissant entrevoir des traitements légers.

Par contre, au rang de ses inconvénients, on peut citer :

- la non-possibilité de gérer la mobilité des objets : Toutes choses égales par ailleurs, le RSSI d'un objet mobile change par principe, ce qui rend *a priori* complexe son identification par ce moyen. Cela dit, c'est un inconvénient moins problématique que ce que l'on pourrait penser. En effet, beaucoup d'objets de la maison connectée sont placés à un emplacement dédié et n'en bougent plus. C'est le cas des ampoules, prises connectées, appareils électroménagers, capteurs de mouvement, serrures de porte, etc. ;
- sa dépendance aux changements d'environnement : Toutes choses égales par ailleurs, le RSSI est volatile au cours du temps dès que l'environnement physique se redessine (habitants qui se déplacent, etc.). Dans un contexte de domicile, c'est un inconvénient rédhibitoire, qu'il faut nécessairement adresser ;
- sa faible granularité : Le RSSI est un indicateur assez grossier. En témoignent au sein de la partie réceptrice d'un émetteur-récepteur CC2531 la précision typique de ±4 dB (sur une plage de 100 dB) ainsi que sa quantification à 1 dBm. Ainsi, deux objets distants de quelques mètres peuvent donner le même RSSI (Nguyen et al., 2011) ;

[10] La partie réception d'un émetteur-récepteur CC2531 effectue par exemple ce calcul de puissance moyenne sur les huit premiers symboles de la trame, c'est-à-dire sur les quatre octets du préambule.

- une corrélation négative entre le RSSI et les conditions de température et d'humidité (Baazaoui et al., 2022). Celle-ci reste cela dit raisonnable. Je n'ai pas trouvé d'étude sur les effets du vieillissement de la sonde et de l'objet.

Des compléments sur le RSSI seront donnés à l'occasion du chapitre sur la réalisation d'un jeu de données Zigbee.

L'utilisation de la densité spectrale de puissance (PSD), est moins courante ; explorée dans (Galtier et al., 2020), elle présente quelques points communs avec le RSSI, étant liée comme lui à la puissance des signaux : la disponibilité sur la plupart des protocoles, la non-possibilité de gérer la mobilité des objets et la dépendance aux changements d'environnement. Par contre, le travail requis pour la capture des données (via une SDR) et l'extraction des attributs est à l'image de celui des attributs radiométriques, conséquent.

2.6.2 Algorithmes pour les empreintes d'objets

L'extraction des attributs effectuée, il est possible de constituer l'empreinte de l'objet. Le but est par la suite d'établir des algorithmes permettant d'identifier les objets et de détecter ceux qui sont illégitimes. Deux grands types d'algorithmes différents ont été recensés : ceux à liste blanche et ceux basés sur de l'apprentissage non supervisé.

2.6.2.1 Algorithmes à liste blanche

Algorithmes à liste blanche basés sur de la classification – Avec ces algorithmes supervisés, on extrait pour tous les objets légitimes, repérés chacun par un identifiant (ID), une série d'attributs, labellisée par l'ID. Un apprentissage est réalisé sur ces données d'entraînement afin de générer un classificateur dont les classes sont l'ensemble des ID. Ensuite, pendant la détection d'intrusion, quand un objet apparait, ses attributs sont extraits et nourrissent le classificateur. Si celui-ci peut affecter une classe, l'objet est considéré comme légitime et peut être repéré par son ID. Dans le cas contraire, l'objet est considéré illégitime et une alarme est levée. Cette technique est à l'œuvre dans la solution (Brik et al., 2008) dont les auteurs disent identifier 130 cartes Wi-Fi du même type. Elle produit en général des résultats avec de très bonnes métriques de détection. Son inconvénient principal est celui de l'apprentissage supervisé, à savoir la création d'un jeu de données labellisées sur lequel entraînement, validation et test doivent ensuite être réalisés.

Algorithmes à liste blanche basés sur des mesures de similarité – Alternativement, si les empreintes des objets peuvent être représentées par un vecteur, on peut pour la phase de détection calculer la similarité de l'empreinte d'un objet apparaissant avec les empreintes de chaque objet légitime, stockées dans une base de données. Cette technique est à l'œuvre dans (Galtier et al., 2020). Combinée à la granularité de la PSD, retenue comme attribut par les auteurs, les travaux identifient 20 instances de modules Zigbee du même type et 18 instances de modules BLE du même type.

Les algorithmes à liste blanche supposent de procéder en amont à un enregistrement des empreintes des objets légitimes, et le cas échéant de construire un classificateur. Ces étapes ne sont pas toujours réalisables par avance. De plus, en cas de nouvel objet à intégrer, la liste blanche change et il faut reconstituer les outils afférents. La scalabilité de cette méthode peut dans certains cas poser question.

2.6.2.2 Algorithmes d'apprentissage non-supervisé

Ces algorithmes sont intéressants dans le sens où un nouvel objet voit son empreinte fabriquée puis la détection de sa légitimité effectuée seulement à son arrivée dans le réseau, sans préparatifs amonts. Si le processus d'extraction des attributs est effectivement simple, on peut se passer d'intervention humaine, ce qui est un élément déterminant.

Détection triviale – C'est la méthode à l'œuvre dans (Faria and Cheriton, 2006). Une anomalie est considérée quand la différence entre des RSSI de référence et les RSSI réels (mesurés par plusieurs points d'accès Wi-Fi distribués, l'ensemble constituant un tuple-empreinte) excède un certain seuil. Ce travail ne gère pas les changements d'environnement physique.

Détection par partitionnement (*clustering*) – Quand les objets émettent des messages, les attributs retenus sont extraits (éventuellement à l'issue de traitements statistiques sur des fenêtres temporelles) et pointés dans un espace des attributs, en gardant une trace de l'adresse MAC. Les points successifs de tous les messages sont utilisés pour former des clusters et, sans attaque, il y a autant de clusters que d'objets. Par contre, si deux ou davantage de clusters sont associés à une même adresse MAC, c'est un symptôme d'attaque d'usurpation de type mascarade. L'attaque est détectée certes mais l'inconvénient est qu'on ne peut savoir lequel des deux clusters caractérise l'attaquant. Au contraire, si deux ou plusieurs adresses MAC sont dans le même cluster, c'est un symptôme d'attaque d'usurpation de type sybil (dans ce cas, on peut compléter une liste noire avec les attributs du cluster en question). Le travail décrit dans (Nguyen et al., 2011) utilise ce principe avec une méthode bayésienne non-paramétrique sur un GMM (*Gaussian Mixture Model*). Celle-ci a l'avantage de ne pas devoir indiquer a priori à l'algorithme le nombre de clusters. Ces algorithmes de partitionnement présentent une complexité de calcul supérieure aux approches basées sur des listes blanches.

Détection par algorithmes s'entraînant sur une seule classe – Les changements d'environnement à l'œuvre dans une maison connectée « vivante » empêchent de considérer le RSSI d'un objet mesuré par une sonde comme une valeur fixe, ou comme une distribution gaussienne autour d'une valeur centrale, comme c'est souvent vu (Jokar et al., 2013). Au contraire, dans ce contexte, le RSSI est plutôt à considérer comme des séquences temporelles. Ainsi, les auteurs de (Madani and Vlajic, 2021) ont proposé de suivre les profils de RSSI par un auto-encodeur à réseaux de neurones récurrents. Pendant l'entraînement, à garantir sans attaque, leur système apprend à modéliser les profils « normaux » de RSSI. Pendant la détection, il arrive à reconstruire les profils RSSI appris pendant l'entraînement mais il échoue à le faire pour des profils qu'il n'a pas appris. Il est considéré que les profils mal reconstruits sont symptomatiques d'une anomalie, par exemple d'une attaque par usurpation d'identité. La validation de ce travail se fait malheureusement dans un environnement simplifié, avec notamment seulement trois nœuds (victime, attaquant, sonde).

2.6.3 Synthèse sur les attributs et algorithmes pour empreintes d'objets

Dans le Tableau 2-5, on classe les travaux présentés selon des critères de praticité, de fonctionnalités et de conditions de validation. Par « Validation réaliste », on entend que les conditions de la validation sont proches des conditions d'une maison connectée : il y a au moins cinq nœuds dans le banc d'essai et les environnements changent (typiquement, à cause de la présence de personnes).

	Facilité d'extraction des attributs	Gestion des environnements changeants	Indépendance par rapport à la modulation	Validation réaliste
(Galtier et al., 2020)	✗	✗	✓	✓
(Brik et al., 2008)	✗	✓	✗	✓
(Faria and Cheriton, 2006)	✓	✗	✓	✗
(Nguyen et al., 2011)	✗	✓	✗	✗
(Madani and Vlajic, 2021)	✓	✓	✓	✗

Tableau 2-5. Points forts et manques des travaux étudiés concernant l'authentification et la détection d'usurpation par empreinte

2.7 Conclusion sur l'état de l'art

Au cours de cet état de l'art, j'ai présenté les caractéristiques des environnements de type « maisons connectées » et constaté la fragilité de leurs objets contraints devant des attaques toujours plus nombreuses. L'exposé d'un profil Zigbee particulier a mis en lumière des vulnérabilités sous la forme d'authentifications absente ou défaillante, ce qui favorise les intrusions notamment par usurpation d'identité. Une présentation des différents types d'IDS a été menée pour permettre par la suite de choisir le plus adapté à ce contexte d'attaque. Enfin, un tour d'horizon relatif à la notion d'empreinte d'objet a montré que des authentifications alternatives basées sur des attributs physiques pouvaient compléter des authentifications existantes voire être la première étape d'une authentification lorsque le système en était dépourvu.

Le chapitre suivant aborde la problématique de cette thèse.

Chapitre 3 - Problématique et thèse

3.1 Problématique

L'état de l'art nous a permis de nous rendre compte que l'environnement IoT des maisons connectées présentaient des caractéristiques propres, très différentes de celles de l'IT, et que celles-ci introduisaient de nombreuses vulnérabilités propices à altérer les propriétés de sécurité (confidentialité, intégrité, disponibilité) des systèmes domestiques présents. Je pense que trois grands facteurs peuvent expliquer la situation de la sécurité des maisons connectées. D'abord, le facteur économique, le plus important à mon avis, conduit à des déploiements d'objets contraints présentant peu ou pas de protection *par conception* et n'ayant pas nécessairement de possibilité de mise à jour. Ensuite, le facteur technologique, relatif à la coexistence de nombreux protocoles, augmente la surface d'attaque et dilue les efforts de recherche de solutions de sécurité. Enfin, le facteur humain traduisant le côté non spécialiste des gestionnaires des IoT domestiques incarne davantage un facteur de risque qu'une valeur ajoutée à la sécurité.

Dans mes travaux sur la sécurité, j'ai choisi de me consacrer au sous-ensemble des attaques de communication visant des objets contraints déjà déployés, avec peu d'espoir d'attention ou de mise à jour (cf. 2.3.5). Elles sont plus faciles à mener que les attaques logicielles et matérielles sur les objets mais elles représentent des nuisances intéressantes et des étapes dans des attaques plus complexes.

Dans ce contexte menaçant, nous faisons malgré tout le constat de l'absence de solutions commerciales de sécurité pour les objets contraints présents dans les maisons. Seuls les objets Wi-Fi bénéficient, en fonction de leurs capacités, de l'environnement TCP/IP de la maison et de ses protections matures (e.g., fermeture des ports inutiles). Pourtant, je pense que la diffusion de solutions adaptées aux caractéristiques des IoT des foyers participerait à des maisons connectées plus sûres, donc à une augmentation de la confiance des utilisateurs et offrirait finalement de nouvelles opportunités de marché.

La littérature fait cela dit état d'une production généreuse sur les systèmes de monitoring d'objets et de détection d'intrusions dans les contextes des maisons connectées. Celle-ci amène de nombreuses contributions sur la connaissance des réseaux IoT et sur la façon de les protéger mais force est de constater qu'il n'y a pas encore eu de transfert significatif de la recherche vers l'industrie pour proposer des sortes de pare-feux IoT économiques, universels et conçus pour des consommateurs devant intervenir le moins possible.

Beaucoup d'articles font figurer l'expression *smart home* dans leur titre mais ils ne considèrent pas nécessairement cet environnement dans toutes ses dimensions. Ils sont la plupart du temps techniques, s'intéressant à la détection d'un certain type d'attaques dans une certaine technologie de communication mais ils font abstraction de l'agressivité économique du secteur, de sa pluralité de piles de protocoles et du côté non-expert des utilisateurs. Ainsi ces ressources ne sont pas directement exploitables pour aborder la conception d'un IDS réaliste pour le grand public. Dans mes lectures, j'ai identifié plusieurs biais récurrents qui retardent la réalisation des IDS recherchés :

- **Solutions complexes, non autonomes, non industrialisables :** Les architectures d'IDS proposées sont parfois complexes et non réalistes pour la destination recherchée ; aussi, des solutions d'HIDS proposées réclament des ressources importantes ne correspondant pas à celles qu'on imagine pour un hôte contraint ; le choix de l'apprentissage supervisé revient par ailleurs souvent alors que la labellisation, pénible et chronophage, est souvent propre à chaque contexte ; sinon, pour constituer des listes blanches, un travail d'enregistrement préalable de chaque

objet est nécessaire, mais celui-ci n'est pas compatible avec la non expertise de l'utilisateur ; en outre, les outils de capture type SDR sont assez plébiscités mais leurs performances relatives et le manque de maturité de leurs écosystèmes font s'interroger sur la possibilité d'une adaptation domestique. La validation expérimentale est parfois non réaliste. Pour finir, le coût de certaines des solutions de sécurité évoquées n'est de surcroit pas en phase avec celui, modeste, d'une installation IoT de maison connectée.

- **Couverture disproportionnée du Wi-Fi :** Sur l'ensemble des publications parcourues concernant des IDS, des systèmes de monitoring et des mesures d'exposition de données privées, je remarque une prédominance de la technologie Wi-Fi sur les autres protocoles IoT alors que le nombre de connexions par cette technologie est à peu près identique à celui des objets au sein des WPAN (typiquement : BLE, Zigbee, Z-Wave). L'hypothèse « IoT = Wi-Fi » semble aller de soi dans beaucoup de travaux académiques. J'ai tenté d'expliquer les causes de ce biais d'habitude en 2.1.1.2. Seuls de rares travaux comme RadIoT (Roux et al., 2018), IoTScanner (Siby et al., 2017) ou IoTHound (Anantharaman et al., 2020) gèrent un ensemble de protocoles représentatifs de l'IoT. En outre, cette répartition en faveur du Wi-Fi se retrouve dans la disponibilité des jeux de données, ralentissant le développement des IDS pour les autres technologies.
- **Identités trop faibles :** Les IDS et outils de monitoring étudiés dans la littérature repèrent très souvent les objets qu'ils suivent par des identifiants de couche MAC, parfois qualifié d'identité faible puisque forgeable facilement, à l'inverse des identités fortes basées sur des attributs physiques difficilement imitables (Bezawada et al., 2018). C'est un problème car au sein des attaques de communication que j'ai choisies d'étudier sur le protocole Zigbee, les attaques de couche MAC par usurpation sont nombreuses. Celles-ci sont permises par une absence d'authentification, une authentification incomplète ou une « clé de réseau » obtenue facilement (cf. sections 2.3.3.2 et 2.4.3.2). En fait, la plupart des travaux ne savent qu'identifier des types d'objets, pas des instances, ce qui les oblige à faire confiance à des identifiants de couche MAC. Il est nécessaire d'utiliser des attributs de couche physique pour identifier des instances. Les attributs radiométriques ne sont cependant pas envisageables dans le cadre fixé, vu leur complexité d'extraction. Quant aux attributs dépendants de l'emplacement, tel le RSSI, ils sont plus accessibles mais ils sont sensibles à la mobilité et aux changements d'environnement. Certes, des travaux comme ceux décrits dans (Faria and Cheriton, 2006) et (Jokar et al., 2013) détectent les attaques d'usurpation par l'utilisation du RSSI avec des heuristiques plus ou moins complexes mais cela se fait dans des environnements statiques de laboratoire ne correspondant pas au dynamisme des domiciles.

Ces réflexions induisent plusieurs questions :

1. Peut-on espérer que, pour un consommateur, équiper son IoT domestique d'une solution de sécurité pour objets contraints soit aussi naturel que d'avoir un pare-feu ou un antivirus dans son installation IT ? Est-ce qu'une approche holistique, c'est à dire économique, technologique et humaine, de l'écosystème maison connectée dans la littérature permettrait de dégager les caractéristiques d'un IDS pouvant être industrialisé en vue d'une diffusion commerciale ?
2. Pour tester et comparer des IDS conçus selon les caractéristiques susmentionnées, des jeux de données sont nécessaires. En trouve-t-on pour d'autres technologies que l'hégémonique Wi-Fi ? Est-il possible d'en trouver pour une technologie dédiée à l'IoT, par exemple Zigbee ? Peuvent-ils faire figurer des attributs de

couche physique afin d'étudier la pertinence de ceux-ci dans l'authentification et la détection d'usurpation d'identité ?

3. Les attaques par usurpation d'identité sont nombreuses et ont un haut rendement dommages / complexité de mise en œuvre. Des solutions pour les détecter existent mais sauf exception, elles ne gèrent pas le caractère habité des domiciles à cause duquel l'environnement est toujours redessiné. Dans le cadre de la réalisation d'un IDS économique testé dans un environnement réel, l'utilisation d'algorithmes d'apprentissage automatique exploitant des indicateurs de couche physique facilement disponibles donne-t-elle des métriques de détection satisfaisantes ?

Ces questions peuvent se résumer en : Peut-on protéger les objets contraints des maisons connectées des attaques de communication (comme celles d'usurpation) par des IDS bon marché, universels et ergonomiques ? D'après la revue de littérature que j'ai menée, cette question n'a pas encore été explorée en profondeur.

3.2 Thèse

Les différentes questions ouvertes de la partie 1.2 et celle synthétique de la fin de la section précédente me permettent finalement de proposer la thèse suivante :

> Afin d'augmenter la confiance dans les réseaux IoT des maisons connectées et de favoriser leur expansion, la sécurité de ceux-ci doit être envisagée de façon aussi naturelle que celle des réseaux de l'IT, ce qui est loin d'être le cas actuellement. Je soutiens que l'observation des caractéristiques des domiciles, la disponibilité de matériels bon marché mesurant aussi les caractéristiques physiques des échanges, l'utilisation de corpus numériques et la démocratisation de l'apprentissage automatique favorisent une approche réaliste des solutions de sécurité contribuant ainsi à leur essor.

Cela implique que je démontre les hypothèses suivantes :

- Des considérations globales sur l'écosystème « maison connectée » et les IDS permettent de définir une base d'exigences pertinentes pour la réalisation de solutions de sécurité réalistes et donc diffusables commercialement.
- La mise à disposition de jeux de données complet d'échanges intégrant, en plus des données classiques de couche liaison, des attributs de couche physique simples d'accès comme le RSSI permet de favoriser la synthèse d'IDS pouvant détecter entre autres les attaques d'usurpation d'identité.
- Des algorithmes d'apprentissage automatique appropriés permettent d'exploiter avec de bonnes métriques de détection un indicateur physique bon marché comme le RSSI, même si celui-ci est volatile en présence des changements d'environnement dus aux activités domestiques.

3.3 Approche proposée

Nous proposons une approche en trois parties pour valider la thèse proposée. Ces trois parties font respectivement l'objet des Chapitre 4, Chapitre 5 et Chapitre 6. Elles ont fait l'objet de plusieurs communications nationales et internationales, référencées en 1.3.2.

- D'abord, je rappelle les facteurs économique, technologique et humain qui façonnent la sécurité des maisons connectées puis les exigences induites sur la conception d'IDS dédiés à ce contexte. En croisant celles-ci avec la taxonomie des IDS établie dans l'état de l'art, je propose une liste de caractéristiques pour des IDS « réalistes » de maison connectée, comme des lignes directrices de conception.

- Ensuite, la pile de protocoles Zigbee me servant dorénavant de fil conducteur, méthodologiquement je propose de construire un jeu de données pour cette technologie, mettant en scène un nombre significatif d'objets connectés statiques dans un environnement habité. Je souhaite que ce jeu de données comprenne classiquement des attributs de couche MAC mais aussi des attributs de couche physique, comme le RSSI, afin qu'il puisse servir à concevoir des IDS traitant les cas classiques de détection, assez répandus dans la littérature, mais aussi ceux liés aux problématiques d'authentification par empreintes physiques. Également, ce jeu se doit de comprendre des exemplaires des attaques en relation que l'on veut détecter.
- Enfin, devant la volatilité de l'attribut de couche physique proposé, des algorithmes d'apprentissage profond menant un travail délicat de traitement du signal sont explorés, dans le but de tendre vers les IDS réalistes recherchés, aptes à détecter les attaques par usurpation d'identité.

Chapitre 4 - Conception d'un IDS réaliste pour maison connectée

4.1 Introduction : une approche holistique

Ce chapitre décrit ma première contribution. Celle-ci se propose de donner des lignes directrices pour la conception d'un IDS dédié au domaine IoT particulier qu'est la maison connectée. L'IDS recherché ambitionne de couvrir les attaques de communication, notamment celles par usurpation. Je considère des objets contraints qui souvent ne peuvent être mis à jour et sont déjà déployés dans les domiciles.

Comme je l'ai déjà évoqué, et c'est ce qui fait l'originalité de ma démarche, je pense que la synthèse d'une solution de sécurité ayant une chance d'être diffusée commercialement doit aborder les objets de la maison connectée pas uniquement sous un angle technique, comme c'est souvent le cas dans la littérature, mais sous une approche holistique, c'est à dire à la fois économique, technologique et humaine. Je crois que c'est une condition nécessaire pour parvenir à un IDS « réaliste ».

Ainsi, si les objets sont bon marché, issus de plusieurs piles de protocoles et gérés par des consommateurs, un IDS réaliste doit respectivement 1) envisager des choix matériels et logiciels à coût raisonnable, 2) gérer les piles de communication les plus populaires et 3) être autonome et favoriser une utilisation simple.

Dans ce chapitre prospectif, les différents éléments constitutifs d'un IDS apte à détecter les attaques de communication seront choisis et présentés sous le prisme de ces contraintes. Nous verrons par la suite si une conception conduite de la sorte amène de la valeur ajoutée à la sécurité.

Dans une première partie, je rappelle les trois **facteurs** permettant d'appréhender l'environnement « maison connectée ». De là, en considérant tour à tour chacun d'entre eux, j'en déduis une liste d'**exigences** pour un IDS bien adapté à cet environnement. Puis, dans une seconde partie, en s'aidant notamment de la taxonomie des IDS établie dans l'état de l'art, j'avance les **caractéristiques** que doit présenter l'IDS recherché. Je termine le chapitre par un exemple d'architecture respectant les différentes caractéristiques à observer. Mes travaux ont été conduits sur le protocole Zigbee mais j'ai vérifié la possibilité de leur généralisation à BLE et Wi-Fi.

4.2 Exigences d'un IDS réaliste

Dans la section 2.2.2 consacrée aux défis de la maison connectée, j'ai identifié trois facteurs façonnant ce domaine. La conception d'un IDS bien adapté à cet environnement doit considérer ces facteurs pour en faire des exigences (*requirements*) à respecter.

4.2.1 Exigences induites par le facteur économique

Tout d'abord, le facteur économique montre un marché piloté par le coût, induisant des objets contraints mal sécurisés. Ces objets sont déployés et livrés à eux-mêmes. Leur ajouter un micrologiciel de sécurité n'est pas envisageable en raison de leurs faibles ressources, des mécanismes existants pour empêcher cet ajout et du côté fermé de certaines parties de leur pile de protocoles. De surcroit, le travail d'écriture devrait être décliné pour chaque système d'exploitation et un système de mise à jour ouvert devrait exister, ce qui est rarement le cas. S'il y a ajout d'une solution de sécurité, celle-ci doit prendre en compte ces obstacles et diversités. Par ailleurs, pour déclencher l'achat, à supposer que les bénéfices de la solution soient suffisamment intéressants, le prix doit malgré tout rester en

rapport avec celui des objets ; une centaine d'euros parait un point de repère initial intéressant pour un prototype.

Ainsi :

1. L'IDS recherché ne doit pas interagir avec les objets pour assurer leur sécurité.
2. Le prix de revient matériel d'un prototype de l'IDS recherché (acquisition des signaux radioélectriques, traitement, génération d'alarme) doit rester sous la barre symbolique des 100 euros, hors ingénierie (programmation, etc.).

4.2.2 Exigences induites par le facteur technologique

Ensuite, ce que j'ai appelé facteur technologique correspond à la présence de plusieurs technologies de communication cohabitant dans un environnement dynamique. Une solution de sécurité idéale doit être agnostique des technologies (piles de protocoles), c'est-à-dire qu'elle peut toutes les gérer, même les nouvelles qui apparaissent. Dans les faits, le seul IDS véritablement agnostique que nous avons rencontré est celui décrit dans (Roux et al., 2018), décrit en 2.5.5.1. Sinon, la littérature fait surtout état de solutions s'occupant uniquement du Wi-Fi, selon des biais introduits en 2.1.1.2 ; seuls quelques travaux déjà évoqués comme IoTHound et IoTScanner gèrent simultanément plusieurs protocoles IoT. En outre, les habitants des maisons connectées ne cessent par leurs activités de redessiner l'environnement géométrique où les ondes évoluent. Certains attributs peuvent y être sensibles, la solution de sécurité doit intégrer cet aspect.

Ainsi :

3. L'IDS recherché doit envisager de gérer les protocoles les plus répandus. La conception doit être guidée par la mutualisation des résultats d'une technologie vers les autres. Il doit pouvoir être mis à jour pour ajouter la surveillance de nouveaux protocoles.
4. L'IDS recherché doit gérer les changements d'environnements physiques dus aux activités humaines (mouvement, déplacement de meubles, etc.).

4.2.3 Exigences induites par le facteur humain

Enfin, le dernier facteur à considérer dans l'approche complète de la maison connectée est le facteur humain. Celui-ci montre un utilisateur sans compétence technique particulière au milieu d'objets déjà déployés. Il n'est pas souhaitable d'engager celui-ci dans une installation nécessitant un enregistrement complexe des empreintes d'objets puis de lui proposer un monitoring de ceux-ci par ordinateur, comme le font les solutions IoTHound et IoTScanner. Des réglages subtils de gain ou des recherches de placement optimum de la sonde ne doivent pas être demandés. Idéalement, la maintenance doit être absente.

Ainsi :

5. L'IDS recherché doit être autonome lors de son placement dans un environnement déjà existant.
6. L'IDS recherché doit être d'une manipulation simple pour les utilisateurs.

4.3 Caractéristiques de l'IDS recherché

Dans cette section, j'étudie les caractéristiques et implémentations possibles de l'IDS recherché en vertu de la liste d'exigences présentée dans la section précédente.

4.3.1 Caractéristiques de l'IDS obtenues via la taxonomie des IDS

Lors de cette première étape, je réinvestis la taxonomie des IDS établie en 2.5.4 afin de déterminer les deux premières caractéristiques de la solution de sécurité que je cherche à concevoir. Pour cela, je reprends les critères « Stratégies de placement » et « Méthodes de détection » de la taxonomie en les croisant avec les exigences d'un IDS réaliste.

4.3.1.1 IDS à placement centralisé observant le réseau

Les HIDS, dans des placements distribués ou hybrides, sont inappropriés dans le contexte d'un IDS dédié à la maison connectée. En effet, développer un micrologiciel additionnel de sécurité pour chaque référence d'objet déployé et considérer son processus de mise à jour ne sont pas des tâches envisageables en raison de la quantité de travail, du manque de ressources des objets et de la non-ouverture de certaines parties des piles de protocoles. Les HIDS doivent aussi être évités parce qu'ils présentent un chemin commun pour à la fois les données de la tâche principale et celles relatives à la détection d'intrusion. Ce manque d'indépendance peut légitimement nous faire nous interroger sur la capacité d'un objet victime d'une attaque DoS à la signaler sachant qu'il ne dispose dans ce cas que de ressources de communication réduite.

Au contraire, un NIDS centralisé présente les avantages suivants : En ce qui concerne la phase de détection (ou de « test »), ses ressources peuvent être dimensionnées pour la complexité de l'utilisation des modèles ou algorithmes retenus et pour une prise en compte de la mise à jour, par exemple pour gérer une pile de protocoles supplémentaire ou ajouter la détection de nouvelles attaques ; typiquement un nano-ordinateur mono-carte à une cinquantaine d'euros de type Raspberry 4 devrait faire l'affaire dans les premières phases de développement, cette hypothèse restant à vérifier. L'utilisation d'accélérateurs matériels pour l'apprentissage automatique, un peu plus chers, fait également partie des pistes à explorer. En ce qui concerne les algorithmes de la phase d'entraînement permettant d'établir des modèles, ceux-ci présentent des complexités algorithmiques (temporelle et spatiale) bien plus importantes que celles liées à la simple utilisation des modèles. Pour ma thèse, entraînement et test seront effectués hors ligne, sur un hôte de type station de travail avec usage de CPU (*Central Processing Unit*) uniquement (i.e., pas de processeur graphique (GPU) ni de processeur pour tenseurs (TPU)). Si l'IDS est validé, il y aura lieu de voir si l'obtention du modèle peut être effectuée sur le même hôte que celui qui pratiquera la détection ou s'il faudra envisager par exemple un service cloud fonctionnant à partir d'un jeu de données capturés par l'utilisateur. Enfin, l'unicité de l'implémentation de l'IDS permet des coûts ainsi qu'une complexité maîtrisés pour la conception, la maintenance et l'exploitation. Cet avantage est aussi un inconvénient car il suffit de mettre la solution centralisée en défaut pour ne plus avoir de protection du tout.

4.3.1.2 IDS comportemental

J'opte pour une détection comportementale (en gardant à l'esprit de conserver un faible taux de faux positifs) car :

- Maintenir une base de données significative des signatures d'attaque dans le contexte hétérogène et disparate de l'IoT est irréaliste ;
- L'IDS doit pouvoir détecter les nouvelles attaques ;
- Un objet IoT réalisant la plupart du temps la même tâche bien définie, son comportement régulier se prête bien à une détection comportementale (Doshi et al., 2018).

4.3.2 Autres caractéristiques de l'IDS

Ensuite, toujours en considérant la conception sous contraintes, je dresse les caractéristiques restantes de l'IDS recherché.

4.3.2.1 IDS agnostique, furtif, passif et autonome dans la découverte des réseaux

Pour couvrir l'hétérogénéité des protocoles IoT, l'IDS doit idéalement être agnostique des piles de protocoles utilisées ou tout au moins envisager de reconnaitre les plus populaires. Idéalement, il n'y a pas à lui fournir de liste de canaux, identifiants réseau, etc.

Pour préserver la simplicité, il doit être furtif, c'est-à-dire ne pas avoir à être intégré comme nœud effectif dans le réseau surveillé. On obtient ainsi une indépendance entre l'IDS et les objets surveillés.

Pour obtenir des informations qui seront transformées en attributs, il doit favoriser l'écoute passive[11]. L'approche active, par exemple pratiquée en envoyant des commandes afin de voir comment le système réagit, permet certes d'obtenir des informations inédites (éléments de qualité de service par exemple) mais cela peut perturber le système, lui faire réaliser qu'il est surveillé ou encore créer de la congestion. Il n'en est rien avec une approche passive car celle-ci permet l'acquisition du trafic réel comme s'il n'y avait pas d'élément d'acquisition. Cette capture en continu pose cela dit des questions sur le respect de la vie privée et par là sur la conformation au Règlement Général sur la Protection des Données, en vigueur en Europe. Les trames Zigbee contiennent des adresses avec un format propre à ce protocole et celles-ci restent en interne du réseau. Par contre, on peut trouver au sein des trames Wi-Fi capturées des adresses IP publiques révélant des données potentiellement sensibles. Cette réflexion n'a pas été abordée davantage au cours de cette thèse.

En ce qui concerne la découverte du réseau, je n'ai travaillé que sur Zigbee / IEEE 802.15.4 : L'émission active de trames de type *Beacon Request* sur les canaux 11 à 26 permet de recueillir des trames *Beacon* de la part des routeurs présents sur les canaux effectivement utilisés, ce qui permet de prendre connaissance de ces derniers[12]. C'est le seul cas où l'IDS doit émettre des messages à destination du réseau IoT surveillé. Par la suite, une écoute passive sur le canal occupé désormais connu permet de recueillir les identifiants 16 bits des objets en présence. La passerelle internet se repère facilement au sein des autres objets par son nombre de trames entrantes et sortantes beaucoup plus important. D'abord parce que c'est un routeur mais aussi parce que tous les ordres donnés par un smartphone et les retours d'état associés transitent par elle.

4.3.2.2 IDS basé sur de l'apprentissage non-supervisé

Apprentissage supervisé vs apprentissage non-supervisé – Comme nous l'avons vu en 2.5.5, beaucoup de solutions de sécurité font usage de l'apprentissage supervisé, leur permettant souvent d'obtenir de bonnes métriques de détection. Ces méthodes requièrent d'abord la constitution en situation de jeux de données labellisés, pour les utiliser ensuite dans les phases d'entraînement et validation. Le test, i.e. la phase de détection en situation réelle, peut alors se faire seulement après toutes ces étapes amont, qui sont coûteuses en temps-ingénieur et pénibles, surtout s'il faut répéter l'opération pour d'autres piles de protocoles. On peut trouver des jeux de données labellisés comportant des attaques mais ils ne sont pas toujours transférables à la situation étudiée et il est difficile d'en trouver pour d'autres piles que Wi-Fi. Je pense que l'usage de méthodes d'apprentissage supervisées est

[11] https://slac.stanford.edu/comp/net/wan-mon/passive-vs-active.html

[12] La commande `zbstumbler` du cadriciel offensif *killerbee* automatise ce processus de découverte : https://github.com/riverloopsec/killerbee

incompatible avec le contexte maison connectée qui me préoccupe, principalement pour les raisons de coût en ressources humaines évoquées.

Par opposition, les algorithmes des IDS comportementaux que l'on souhaite construire (pour détecter des attaques, inférer des types d'objets, etc.) ne doivent donc pas nécessiter des jeux de données ayant besoin d'être labellisées. L'apprentissage non-supervisé (*unsupervised learning*) se charge justement de trouver des structures cachées dans des données non labellisées (Géron, 2019). Dans ce domaine, outre les techniques de réduction de dimension (analyse en composantes principales, algorithme de visualisation t-SNE[13], entre autres), on trouve un grand nombre de techniques de partitionnement (*clustering*) telles K-Means, HDBSCAN[14] et les mixtures gaussiennes. Les données capturées, élaborées en attributs, peuvent être placées au fur et à mesure de leur arrivée dans l'espace des attributs et les algorithmes précités les affectent à des clusters. Un nouveau point capturé ayant des attributs tels qu'il ne rentre pas dans un cluster prévu peut témoigner du comportement étrange d'un objet, symptôme possible d'une compromission.

Apprentissage non supervisé de type partitionnement : un exemple – La Figure 4-1 montre ainsi un travail de partitionnement, effectué hors-ligne grâce au jeu de données décrit au Chapitre 5, portant sur les empreintes de 8 objets Zigbee, collectées le 1er juillet 2022 de 2h15 à 2h45 (CEST), période de sommeil pendant laquelle l'environnement physique n'a pas changé (il y aura lieu plus tard de s'intéresser à ce qu'il se passe quand l'environnement est dynamique). Il n'y a pas eu d'attaque pendant cette capture. À chaque émission d'une trame par un objet, un tuple de 4 RSSI fournis par 4 sondes distribuées dans le domicile est obtenu. Les objets ont été discriminés par leur identifiant 16 bits et la période de capture a été divisée en fenêtres temporelles de 10 secondes, durée argumentée dans (Anantharaman et al., 2020). Pour chaque fenêtre, 4 RSSI médians sont calculés par objet et le tuple de ces 4 valeurs constitue son empreinte physique, par exemple (-69, -62, -66, -89) (en dBm). Pour la visualisation en deux dimensions de ces données à quatre dimensions, j'utilise l'algorithme t-SNE[15] implémenté dans la bibliothèque Python *scikit-learn* (Pedregosa et al., 2011) dédiée à l'apprentissage automatique. On voit que les objets forment des clusters d'empreintes bien distincts, ce qui permet de les monitorer : Dans cet environnement statique, si un attaquant venait à se faire passer pour un objet en usurpant son identifiant 16 bits (attaque de type mascarade), il apparaitrait un deuxième cluster pour ce même identifiant 16 bits. Cela permet de détecter l'attaque mais pas de la caractériser car l'empreinte de l'attaquant est aussi vraisemblable que celle de la victime[16], ce qui est problématique. En effet, le système n'est pas prévu pour apprendre les empreintes légitimes de chaque objet. Au contraire, lors d'une attaque sybil où un attaquant prend plusieurs identités, il y aurait plusieurs identifiants 16 bits au sein d'un même cluster. On peut classer cette empreinte d'attaquant dans une liste noire, constituant par là un début de mitigation.

[13] t-SNE : *t-distributed Stochastic Neighbor Embedding*.
[14] HDBSCAN : *Hierarchical Density-Based Spatial Clustering of Applications with Noise*
[15] Les paramètres utilisés pour l'instanciation de la classe `TSNE` sont, pour ce graphe : `perplexity=178, init='random', learning_rate=200.0`.
[16] Sauf à considérer que les attaques sont statistiquement plus rares que les non-attaques.

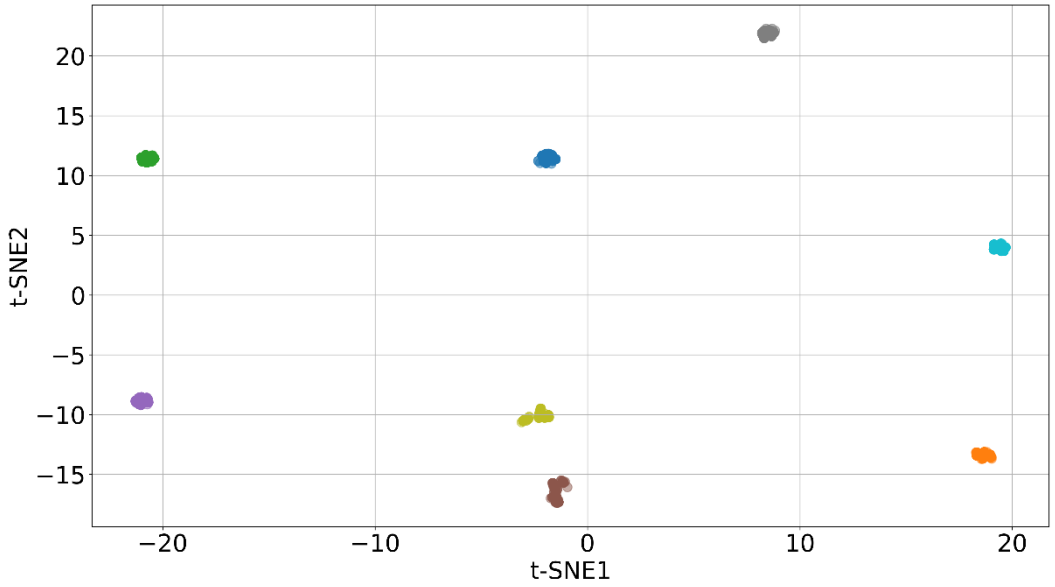

Figure 4-1. Partitionnement d'empreintes formées de 4 RSSI par objet (8 objets Zigbee, environnement statique, pas d'attaque)

Apprentissage non supervisé avec entraînement sur données d'une seule classe – Une autre série d'algorithmes non supervisés comme *Isolation Forest*, *Local Outlier Factor* ou *One-class Support Vector Machine* sont spécialisés dans la détection d'anomalies. Il est nécessaire de leur fournir des données d'une seule classe, la plupart du temps celle correspondant à une activité légitime (i.e., sans attaque), ce qui revêt un aspect très pratique ; évidemment, ces algorithmes sont à éviter en environnement majoritairement hostile. Un modèle permettant d'évaluer l'appartenance à la classe est généré à l'aide de ces données légitimes. Par la suite, en test, les attributs d'un nouveau point capturé nourrissent le modèle et celui-ci statue de la légitimité ou de l'illégitimité de ce point. C'est également ce principe d'entraînement sur une seule classe qui est utilisé avec les auto-encodeurs à réseau de neurones servant pour la détection d'anomalies dans (Roux et al., 2018) ou (Madani and Vlajic, 2021), déjà évoqués. Cet apprentissage non-supervisé est parfois qualifié à juste titre d'apprentissage auto-supervisé (*self-supervised learning*). Par la simplicité de la constitution du jeu d'entraînement et par le fait qu'il puisse fournir l'identifiant de l'objet victime d'une usurpation, ce type d'algorithme me semble le plus adapté à l'IDS recherché.

4.3.2.3 IDS autonome dans la réalisation d'empreinte

Dans le contexte qui m'intéresse, l'extraction d'attributs pour la constitution d'empreinte, que ce soit pour établir une liste blanche et/ou pour reconnaitre cette empreinte pendant la phase de test, doit pouvoir être réalisé rapidement et de façon automatique. Aucune intervention humaine n'est envisageable. Ainsi, seuls des attributs faciles à extraire me paraissent viables. Ce parti-pris éloigne de fait l'utilisation d'attributs radiométriques (ou de complexité d'extraction similaire) qui m'ont semblé à chaque fois nécessiter un travail humain non négligeable.

4.3.2.4 IDS utilisant des attributs de couche MAC

Un premier argument qui pousse à utiliser les informations de couche MAC est que la disponibilité des informations de couche PHY dépend trop de la technologie, mise à part celle du RSSI. Par ailleurs, les couches au-dessus de MAC sont en général chiffrées et donnent de fait plutôt accès à des métadonnées.

Afin de prendre en compte l'exigence de gestion de multiples piles de communication, il me semble intéressant de recenser des attributs communs aux différentes piles présentes et pertinents en apparence. Ainsi, si une solution efficace est établie pour une technologie, sa mutualisation aux autres technologies permettra des économies. C'est au niveau de la couche MAC que l'on peut trouver des informations communes et donc des attributs communs. C'est l'idée suivie par les travaux de (Anantharaman et al., 2020) menés sur BLE, Wi-Fi et Zigbee. Au sein de fenêtres temporelles de quelques secondes, les auteurs ont généré des statistiques portant sur le temps inter-trames, la longueur de charge utile, la longueur de trame, l'entropie de charge utile et l'entropie de trame. Ils ont d'abord développé la solution pour Wi-Fi et l'ont adapté à BLE et Zigbee en optimisant la longueur de la fenêtre temporelle. Celle-ci est fixée pour chaque protocole par le respect de la meilleure justesse possible dans la reconnaissance de type.

4.3.2.5 IDS avec sondes bon marché à données démodulées

Deux options peuvent être considérées concernant l'acquisition des informations par une sonde pour en faire des attributs alimentant des modèles : les radios logicielles (SDR) et les émetteurs-récepteurs dédiés à une technologie utilisés en sonde en écoute passive :

SDR – Les SDR sont des émetteurs-récepteurs où seule la couche physique est implémentée de façon matérielle. En mode acquisition, elles capturent moyennant un changement de fréquence les signaux électromagnétiques sur de larges bandes, de manière agnostique, tandis que le travail de décodage des couches supérieures s'exécute sur un processeur. À première vue, la SDR semble l'outil idéal, supportant potentiellement autant de protocoles que nécessaire, même les nouveaux par une mise à jour logicielle. Elles donnent en plus accès à des caractéristiques de couche PHY auxquelles les récepteurs dédiés n'ont pas accès, permettant de les envisager dans des problématiques d'authentification en mesurant des attributs radiométriques. Malheureusement, le domaine des SDR n'est pas encore mature et beaucoup de temps-ingénieur doit être consacré à programmer des démodulateurs et des décodeurs puis à régler de façon optimum un certain nombre de gains, rendant l'exploitation par un consommateur irréaliste. En outre, le volume de données brutes qu'elles collectent au format (I, Q) représente plusieurs mégaoctets par seconde de capture (selon la fréquence d'échantillonnage et la résolution de la SDR), ce qui n'est pas compatible avec un processeur d'analyse à ressources et coût raisonnables, ni avec la conservation de ces données. Par ailleurs, le prix d'un tel outil atteint lui-même plusieurs centaines d'euros, faisant définitivement s'éloigner la perspective d'un IDS respectant les exigences économiques précitées. Je ne pense pas que les SDR puissent rentrer dans le cadre précis de la synthèse d'un IDS dédié au contexte maison connectée.

Sondes bon marché – On peut plutôt utiliser des puces émetteurs-récepteurs en récepteurs uniquement qui démodulent les signaux d'une technologie. Ces sondes passives (*sniffers*) peuvent se présenter avec une interface USB, ce qui permet d'effectuer les premiers développements de capture sur une station de travail ou un nano-ordinateur. Les sondes sont bon marché, disponibles sur étagère (COTS), de faible consommation et fonctionnent immédiatement. Par exemple, un CC2531EMK dédié à Zigbee / IEEE 802.15.4 coûte moins d'une dizaine d'euros, livré avec une antenne externe d'environ 8 cm. Bénéficiant d'une sensibilité de -97 dBm (0.2 milliardièmes de Watt), il peut couvrir, à condition qu'il soit bien placé, un réseau Zigbee d'une maison de 100 m^2 sur deux étages. La SDR BladeRF 2.0 (avec antennes) que j'ai utilisé en début de thèse couvre plutôt une pièce de cette maison. En outre, on peut désormais remplacer un ensemble de plusieurs sondes par une puce gérant plusieurs protocoles, à l'image du nRF52840 qui gèrent sept piles (dont BLE et Zigbee) pour un coût inférieur à dix euros. D'autres puces comme les CC2652R, destinées à équiper des passerelles IoT multi-protocoles (BLE, Wi-Fi, Zigbee), sont également disponibles mais elles sont plus chères (environ 60 euros). L'intérêt des sondes

passives est de fournir directement les données démodulées sous forme de trames de couche MAC, avec une grande fiabilité et un faible volume de données. Une exploitation temps réel ou hors-ligne avec Wireshark ou via des bibliothèques Python (telles Scapy[17] ou Pyshark[18]) sont immédiates. Par ailleurs, l'accès au RSSI (grandeur de couche PHY) est en général accessible depuis un champ particulier de la trame, ce qui permet de garder l'espoir d'une possibilité d'identification au niveau instance si on utilise cette solution plutôt qu'une SDR. Finalement, même si ajouter la prise en charge d'une nouvelle technologie au cours de la vie de l'IDS oblige à ajouter physiquement un nouveau *sniffer*, cette option me semble la meilleure solution pour la partie acquisition de l'IDS recherché.

4.3.2.6 IDS ergonomique, en phase avec le profil « consommateur »

Je dresse ci-dessous une liste des dernières caractéristiques nécessaires de l'IDS, en lien avec le profil non-expert de l'utilisateur.

Installation – L'IDS doit être facile à installer, il ne s'agit pas de faire venir un expert dans le domicile, vu le contexte économique. Éventuellement, des indications sur l'emplacement le plus favorable pour la détection peuvent être fournies dans un mode d'emploi. Les réglages initiaux subtiles doivent idéalement être absents.

Usage – L'IDS doit rester simple à l'usage. S'il est basé sur un entraînement avec des situations à une classe, on peut imaginer un bouton « LEARN » (apprendre) sur lequel l'utilisateur appuie quand il est confiant sur l'absence d'attaque dans les quelques heures d'apprentissage à venir.

Pour les deux items précédents, le **caractère autonome** de l'IDS est important. Ce point a été abordé à plusieurs reprises (cf. 4.3.2.1 et 4.3.2.3).

Interface homme-machine – Le smartphone me semble la seule interface homme-machine possible dans ce contexte d'utilisateur-consommateur, soit via une interface web, soit via une application.

Alarme – En cas d'attaque détectée, une notification d'intrusion doit être envoyée à l'utilisateur. Étant donné que j'ai considéré pour l'instant un IDS totalement indépendant du réseau à surveiller, cette notification est à envoyer par un autre canal que l'application-compagnon qui gère le réseau. Cela peut prendre la forme d'un SMS ou d'une notification dans une application dédiée à l'IDS. Ensuite, il y a lieu de s'interroger sur le type d'information que le système est capable de remonter : Signale-t-il sans plus de détail qu'une attaque est en cours ? Peut-il préciser l'objet victime ? Est-il en mesure d'annoncer de quel type d'attaque il s'agit ?

Mitigation – Quand une attaque est détectée, quelles actions simples de mitigation peuvent être envisagées : Inciter l'utilisateur à mettre l'objet victime en quarantaine ? Chercher des traces de l'attaquant dans le voisinage ?

Facteur de forme – Pour des raisons pratiques, l'IDS doit être sous forme de *box* compacte, si possible en une pièce, à déposer à un endroit central. Dans un premier temps, la viabilité de la solution sera donc étudiée avec une seule sonde connectée. Je verrai ensuite si plusieurs sondes sont nécessaires pour des besoins de couverture du domicile ou des raisons de corrélation nécessaire des informations. Je suis conscients de ne pas avoir mené une véritable étude d'industrialisation, hors de mes compétences, mais les solutions présentées, même si elles reposent sur de l'assemblage de composants sur étagère conduisent à un encombrement déjà modeste qu'une étude d'industrialisation pourra réduire davantage.

[17] https://scapy.net/
[18] https://pypi.org/project/pyshark/

Changements d'environnements – Les habitants, par leurs mouvements et leur réorganisation de l'espace modifient le comportement des ondes électromagnétiques. Des attributs comme le RSSI y sont sensibles. S'il est utilisé, il faut gérer la volatilité de cet indicateur.

Confiance – L'utilisateur ne doit pas être importuné par des successions de faux-positifs qui feront perdre sa confiance en l'IDS. Les faux-négatifs ne sont pas souhaitables mais dans ce cadre, ils sont peut-être préférables.

Coût – En vertu des différentes indications financières fournies, il apparait que le coût d'un prototype, hors travail d'ingénierie, se situe sous la centaine d'euros, se répartissant de manières égales entre le bloc de traitement et les sondes dédiées.

4.4 *Exemple d'architecture*

À titre d'illustration, on donne Figure 4-2 un exemple d'architecture d'IDS domestique respectant au maximum les caractéristiques énoncées dans cette contribution. Celui-ci est non supervisé et s'entraîne sur des données d'une seule classe. Selon les attributs retenus, il peut servir par exemple à détecter des appareils pratiquant un DoS ou à détecter des attaques d'usurpation. Dans ce dernier cas, discriminer des identités suppose des données de couche physique. Le RSSI en est une, heureusement disponible depuis un champ de couche MAC dans la plupart des récepteurs dédiés à une technologie.

4.4.1 Entraînement

Le principe retenu est, pour chaque technologie couverte (Wi-Fi et Zigbee sur Figure 4-2), d'entraîner le système sur une unique classe, logiquement celle où les échanges sont légitimes. Un « mode entraînement » peut être programmé pour durer quelques heures après que le bouton « LEARN » ait été pressé. Pendant tout ce temps, l'IDS réalise l'acquisition des données Wi-Fi et Zigbee démodulées par les sondes et contribue après extraction des attributs par objet à élaborer un modèle ou une empreinte du comportement légitime de chaque objet. Le sous-bloc « Nettoyage » filtre les données aberrantes. Quant au sous-bloc « Standardisation », il permet d'obtenir des attributs standardisés évoluant dans des ordres de grandeurs proches, ce qui est souvent requis pour un bon fonctionnement des algorithmes d'apprentissage automatique sans réseau de neurones ; cela assure en même temps des grandeurs d'amplitude raisonnable en entrée des fonctions d'activation des algorithmes à réseaux de neurones. Pour une bonne détection des anomalies, la phase d'apprentissage doit se faire dans un contexte ressemblant le plus possible à des situations « normales » : évolution de l'environnement physique (par déplacement de personnes ou de meubles), activité des objets, positions des sondes et objets, puissance d'émission des objets doivent être représentatives de la vie quotidienne de la maison.

4.4.2 Détection

Quand l'entraînement est terminé, le système repasse en « mode détection ». Il utilise toujours les blocs « Acquisition », « Nettoyage », « Extraction » et « Standardisation » mais il peut y avoir bien-sûr des attributs liés à des attaques durant ce mode. Un bloc de détection d'anomalies paramétré par les modèles ou empreintes établies à l'étape précédente permet de découvrir si le comportement de chaque objet est normal et si ce n'est pas le cas, une alarme est générée.

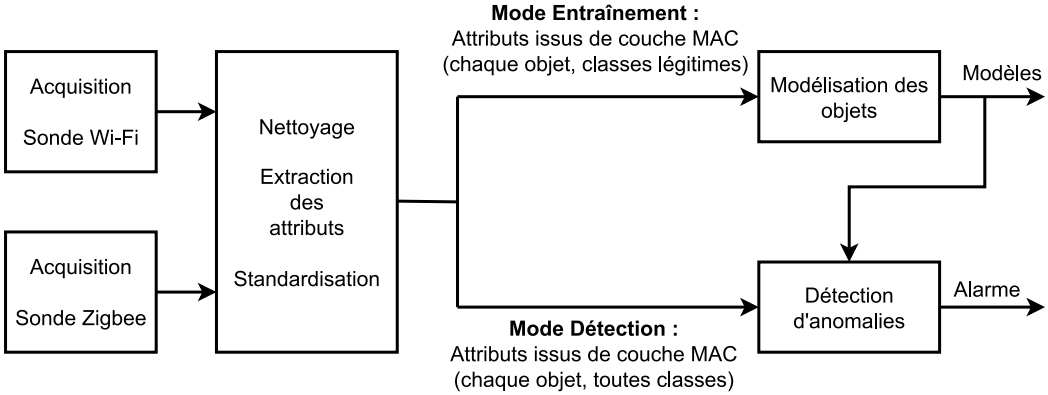

Figure 4-2. Exemple d'architecture d'IDS comportemental à apprentissage non-supervisé (entraînement sur des données d'une seule classe)

4.5 Conclusion

Dans ce chapitre, j'ai proposé une approche de la maison connectée sous un angle beaucoup plus ouvert que celui de nombreux travaux, purement techniques. Recensant les facteurs économique, technologique et humain de la sécurité de cet environnement, j'ai imaginé par ricochet quelles exigences cela pouvait induire sur la conception d'un IDS. Un représentant respectant ces exigences serait de fait bien adapté à l'environnement, et par-là diffusable comme le sont les anti-virus, pare-feux et autres solutions de protection dans le monde de l'IT.

Ainsi, le faible coût des objets doit inciter à concevoir des IDS peu chers prenant en compte les faibles capacités des objets qu'ils doivent protéger. Ce critère économique conditionne alors inhabituellement des choix très importants de l'IDS comme la méthode d'apprentissage, non supervisée plutôt que supervisée. Les faibles capacités des objets ainsi que leur variété impliquent également la réalisation d'un système de protection passif.

La prise en compte de la multiplicité des protocoles au sein d'une maison, souvent négligée au sein de la littérature des IDS, davantage encline au seul Wi-Fi, m'a amené à réfléchir sur les moyens d'acquisition des données. Ce sont encore des critères de coût et de praticité qui ont permis d'élire l'option de sondes bon marché plutôt que des radios logicielles.

Enfin, une large place a été faite à l'étude des habitudes et des capacités des utilisateurs. Celles-ci ne sont pas celles d'un administrateur et elles orientent la conception de l'IDS vers des critères d'autonomie et de simplicité d'utilisation. À la fin, le coût matériel d'un prototype ne dépasse pas cent euros, c'est-à-dire le prix raisonnable de quelques ampoules connectées.

Avec cette contribution, je pense avoir apporté une vision originale sur le contexte de la maison connecté et par là sur la conception des potentiels IDS qui pourraient la protéger. Je croie que ce travail peut contribuer à diffuser des IDS domestiques. Pour que cette première démarche de réflexion ne reste pas formelle, il est nécessaire d'élaborer des IDS selon les contraintes présentées dans ce chapitre et d'analyser leurs résultats. Méthodologiquement, je propose de le faire en utilisant d'abord un outil du type jeu de données (*dataset*) qui procurera de la souplesse dans l'étude et des possibilités de comparaisons d'IDS. Respectant cette orientation, le chapitre suivant est consacré à la conception d'un jeu données Zigbee pour la maison connectée.

Chapitre 5 - Réalisation d'un jeu de données Zigbee

5.1 Introduction : motivations pour un jeu de données Zigbee

Les chapitres précédents ont montré l'intérêt en termes de sécurité de l'intégration d'IDS dans les environnements fragiles de type maison connectée. Dans le cadre de la conception de ces systèmes, des jeux de données représentent un outil obligatoire. D'abord pour concevoir et évaluer ces IDS avec le maximum de praticité, puis, une fois ceux-ci réalisés, pour comparer leurs performances avec celles des solutions existantes.

Souhaitant expérimenter l'utilisation d'un indicateur physique facilement accessible dans le cadre de la détection des attaques par usurpation, Je me suis mis en recherche d'un jeu de données Zigbee réaliste, c'est-à-dire capturé dans une maison habitée et faisant figurer des phases bénignes et des phases d'attaques, labellisées. Mais, de même que beaucoup d'articles de recherche mentionnant « smart home » dans leur titre ne considèrent souvent que les seules piles Wi-Fi, négligeant par-là le côté hétérogène de ce contexte, les jeux de données de qualité se réclamant de l'IoT et de la maison connectée ne concernent également souvent que le Wi-Fi, bien que, rappelons-le, cette pile de protocoles ne représente qu'un tiers des connexions actives (cf. 2.1.1.2).

Devant la non disponibilité d'un tel outil, j'ai décidé de construire le mien puis de le mettre à disposition de la communauté, d'abord dans un esprit de science reproductible sur les solutions que je proposerai, puis pour inciter les chercheurs en cybersécurité à développer des IDS dédiées à des IoT implémentant d'autres piles que Wi-Fi. Mon jeu de données, baptisé ZBDS2023, est le reflet de 10 jours d'échanges entre 10 objets Zigbee liés à des fonctionnalités d'éclairage, répartis dans une maison habitée, donc à environnement physique changeant[19]. L'élaboration et les caractéristiques du jeu de données sont complètement décrites dans ce chapitre. De même, les attaques de communication injectées sont toutes décrites, référencées et labellisées. Chaque trame émise par un objet est capturée par une à quatre sondes bon marché dédiées à la pile Zigbee.

Ce chapitre commence par un bilan de la disponibilité des jeux de données Zigbee. Ensuite, il poursuit l'exposé de l'attribut RSSI entrepris dans l'état de l'art en raison de la facilité d'accès ce cette donnée de couche physique. Les caractéristiques du jeu de données proposé sont ensuite exposées. Puis, le banc d'essai ayant servi à la capture est présenté à travers la présentation des sondes et des objets utilisés. La liste des attaques menées vient par la suite. Finalement, un cas d'IDS naïf basé sur l'exploitation du RSSI est réalisé avant de conclure.

5.2 Jeux de données Zigbee existants

Dans la revue de littérature (Jmila et al., 2022) sur la classification des objets, on apprend que seulement 5% des papiers considérés partagent leur jeux de données.

En outre, en cherchant des jeux de données IoT de qualité, on trouve surtout des jeux de données Wi-Fi. Par exemple, les auteurs de (Alrawi et al., 2019) proposent un jeu de données dit « IoT » populaire, complétement annoté, incluant des attaques et permettant l'évaluation fine de la sécurité de 45 appareils et de leurs écosystèmes (cf. 2.2.1). Ce jeu

[19] Nous n'avons pas envisagé l'utilisation de simulateurs tels Cooja (Wallgren et al., 2013) ou Avrora (Stelte and Rodosek, 2013), ceux-ci ne permettant pas de prendre en compte les évolutions du canal physique dues aux changements d'environnement, typiques d'un domicile.

est malheureusement exclusivement Wi-Fi, à l'image de celui décrit dans (Ren et al., 2019) permettant d'évaluer comment les objets exposent des données privées vers Internet.

Au contraire, des travaux comme (Anantharaman et al., 2020) ou (Acar et al., 2020) considèrent l'hétérogénéité de l'IoT en manipulant de grandes quantités de données BLE, Wi-Fi et Zigbee afin de construire des systèmes de monitoring et des IDS. Malheureusement, leurs jeux de données n'ont pas été rendus publics empêchant de reproduire simplement des parties de leurs expériences.

Dans (Dadkhah et al., 2022), les auteurs ont rendu public un jeu de données dépassant le cadre du Wi-Fi en implémentant un banc d'essai de 60 objets Wi-Fi, Zigbee et Z-Wave. En outre, ils ont collecté systématiquement des trames de chaque objet dans plusieurs étapes comme la mise sous tension, l'association et l'interaction avec d'autres objets. Ainsi, ils tentent de reproduire une collection de comportements élémentaires survenant dans une maison connectée. Ils ont aussi capturé plusieurs associations IEEE 802.15.4 qui m'ont permis de récupérer certaines de leurs clés de réseau (cf. 2.4.3.2), montrant comment ce procédé est peu sûr. Cependant des attaques actives n'ont été menées une fois de plus que sur les objets Wi-Fi, empêchant d'utiliser leur jeu de données pour construire un outil de monitoring ou un IDS pour Zigbee.

Il est bon de préciser que les travaux cités jusqu'à maintenant dans cette section ne font figurer qu'une instance d'objet par type d'objets envisagés, un biais que nous avons déjà évoqué en 2.5.5.3.

Les travaux décrits dans (Galtier et al., 2020), déjà évoqués en 2.6.1.2, sont capables de différentier plusieurs instances du même type d'objets. Les auteurs ont intelligemment publié leurs jeux de données. Cela dit, les attributs retenus et leurs modalités d'extraction ne sont pas compatibles avec les IDS simples que je souhaite construire.

Un autre projet ambitieux est décrit dans (Duque et al., 2021). Couvrant grâce à des SDR des objets BLE et Zigbee, ce jeu de données fait figurer des données (I, Q) et les trames relatives démodulées. Ce travail est orienté sur les empreintes et la localisation des objets mais il ne fait pas mention d'attaques conduites.

Pour compenser la rareté des jeux de données, remplacer un déploiement peu pratique de nombreux objets ou éviter de se retrouver en dehors de la réglementation sur le respect de la vie privée (par défaut d'anonymisation par exemple), la génération automatique de jeux de données est une option à considérer. Les travaux exposés dans (Cordero et al., 2021) ne sont pas spécifiquement orientés IoT mais proposent d'injecter des attaques sur un arrière-plan de fichiers PCAP de trafic calme, afin de constituer des jeux de données labelisés. Les travaux décrits dans (Shahid et al., 2020) s'appuient quant à eux sur des réseaux antagonistes génératifs (*Generative Adversarial Networks*, GAN) pour générer du trafic IoT avec des caractéristiques semblables à du vrai trafic IoT. Ces thématiques de génération de jeux de données constituent un sujet de recherche actif dans la sécurité, elles sont à explorer sans délai pour les protocoles les plus représentatifs de l'IoT.

5.3 RSSI pour l'identification et la détection des usurpations

Je complète ici la présentation du RSSI commencée en 2.6.1.2. Celui-ci est un attribut physique, il est donc candidat à la génération d'empreinte pour l'identification des objets et la détection des attaques d'usurpation. Il est en outre le seul de cette nature accessible depuis la couche MAC de sondes passives bon marché. Dans cette section, afin d'orienter la conception de mon jeu de données, j'étudie ce que l'on peut espérer construire avec le RSSI mais également les limitations de cet attribut.

5.3.1 En espace libre

Une attaque par usurpation d'identité a lieu lorsqu'un attaquant prend l'identité d'un nœud légitime pour mener des actions inappropriées sur un nœud victime. Pour expliquer le principe de la détection d'usurpation par le RSSI, je considère dans un premier temps que nous sommes en espace libre (i.e., sans obstacle). Dans l'extrait de réseau Zigbee de la Figure 5-1, l'identité d'un objet est assurée par l'identifiant 16 bits de couche MAC IEEE 802.15.4. Pour économiser de l'énergie, le capteur de mouvement d'identifiant 0x0A12 est la plupart du temps en mode sommeil. Il se réveille quand il détecte un mouvement ou sinon 12 fois par minute. Il demande alors à son nœud parent 0x7C77, ici une ampoule connectée, si celui-ci a reçu des données à son attention pendant qu'il dormait. 0x0A12 formule cette demande avec une trame IEEE 802.15.4 dite de *Data Request*.

Dans une attaque par usurpation de type mascarade (*Masquerade*), un attaquant peut usurper l'identité de 0x0A12 en forgeant à son tour des trames *Data Request* avec cet identifiant source. Celles-ci étant formées correctement, la victime se comportera normalement à cette sollicitation. Si cette requête se répète plusieurs centaines de fois par minute (mode inondation), la victime 0x7C77 va se mettre à manquer de réactivité dans le traitement des ordres d'allumage/extinction et le réseau risquera d'être congestionné, deux formes de dénis de service, le premier pouvant être un point-étape vers des attaques dangereuses sur les couches hautes (cf. 2.4.3.2).

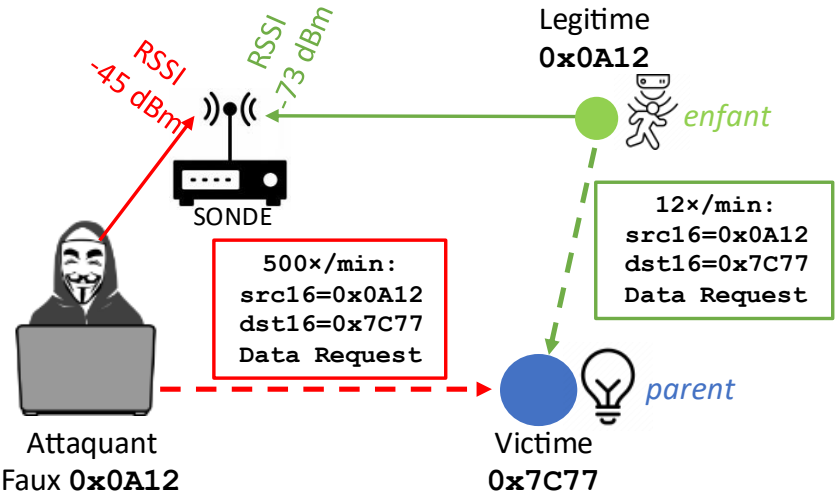

Figure 5-1. Usurpation d'identité dans un réseau Zigbee et détection (principe)

Pour détecter cette usurpation, on peut placer une sonde d'un IDS dans l'environnement, comme sur la Figure 5-1. Pour la suite, on considère qu'attaquant, objet et sonde sont chacun à leur place dédiée et qu'attaquant et objet émettent avec leur propre puissance d'émission, mais que celle-ci reste constante, une hypothèse raisonnable. L'IDS voit arriver pour l'identifiant 0x0A12 des trames avec une certaine valeur de RSSI[20], qu'il a pu associer préalablement dans une liste (constituée pendant une phase sans attaque), à l'identifiant 0x0A12. En cas d'attaque d'usurpation, il voit aussi arriver des trames avec un RSSI

[20] Même en environnement statique sans attaque, le RSSI n'est pas constant et présente une distribution, souvent considérée comme gaussienne dans la littérature. On devrait donc dire : « L'IDS voit arriver pour l'identifiant 0x0A12 des trames avec un RSSI appartenant à une plage de valeurs plus ou moins resserrées autour de la moyenne du RSSI. »

différent de celui qu'il a appris pour 0x0A12, ce qui permet de générer une alarme. Ce principe n'est pas infaillible car il existe des combinaisons (puissance d'émission, distance à la sonde) pour l'attaquant qui pourraient produire le même RSSI sur la sonde que l'objet légitime. Cet « échappement de la détection » (*detection evasion*), fortuit ou recherché, serait à l'origine de faux négatifs. Il y a lieu d'évaluer ce risque et de voir si celui-ci est acceptable.

Sinon, dans une problématique d'identification non basée sur les adresses MAC ou équivalent, les identifiants doivent être uniques par objet. Une seule valeur de RSSI ne peut jouer ce rôle car, à supposer que tous les objets aient la même puissance d'émission, tous ceux situés sur un même cercle ayant pour centre la sonde ont théoriquement le même RSSI. L'association d'un RSSI à chaque objet n'est pas bijective. La faible granularité de cet attribut n'améliore pas les choses, plusieurs objets proches donnant également le même RSSI.

5.3.2 En intérieur, en environnement statique

En intérieur, dans un environnement avec obstacles statiques, la formule (2-6) donnant l'expression du RSSI en espace libre (cf. 2.6.1.2 page 58) n'est plus valable car absorption, réflexion, diffraction et dispersion dues aux obstacles génèrent des chemins multiples pour les ondes, avec des interférences inter-symboles et un signal de plus faible qualité à l'arrivée (Farahani, 2008). Même si les obstacles peuvent contribuer à mieux distribuer des valeurs différentes de RSSI pour tous les objets et à complexifier le travail d'imitation par un attaquant, l'association d'un RSSI à chaque objet n'est toujours pas bijective.

5.3.3 Amélioration de l'identification et de la détection d'usurpation par l'utilisation de plusieurs sondes

Pour progresser dans la problématique d'identification/détection d'usurpation et rendre également moins probable l'échappement de détection par l'attaquant, on peut distribuer plusieurs sondes dans le domicile et caractériser un objet émetteur par un tuple de RSSI plutôt que par un seul RSSI, obtenant une empreinte plus robuste (dans l'esprit de ce qui est fait dans la localisation par trilatération). Cette idée n'est pas neuve, elle a été par exemple explorée en profondeur dans (Faria and Cheriton, 2006) pour détecter avec succès les usurpations d'identité (cf. 2.6.2.2) en Wi-Fi, dans un environnement statique. Dans le travail de regroupement de la Figure 4-1, obtenue en environnement statique, j'avais utilisé pour caractériser chaque objet un tuple-empreinte de quatre RSSI issus de sondes distribuées dans le domicile. J'avais obtenu une empreinte unique par objet, une étape intéressante dans la problématique d'identité forte.

Les IDS évoqués à l'instant nécessitent plusieurs sondes pour mesurer les différents RSSI. Ils requièrent de ce fait une nouvelle architecture d'acquisition et un regroupement des informations par trame, chacune donnant lieu à plusieurs captures. Cela augmentera nécessairement la complexité et le coût de la solution de sécurité que je cherche à promouvoir. Sans présumer de ces difficultés techniques et financières, il me parait malgré tout important que mon jeu de données capture les trames depuis plusieurs emplacements pour offrir une de la souplesse dans la synthèse d'IDS.

5.3.4 Nécessité de capture dans un environnement réaliste

Dans les travaux sur l'usurpation d'identité décrits dans (Faria and Cheriton, 2006) et dans beaucoup d'autres, les mesures de validation se déploient dans des environnements statiques peu en rapport avec ceux sans cesse redessinés d'une maison connectée vivante où l'évolution temporelle du RSSI semble stochastique et complexe à appréhender. Zigbee

étant une technologie pour les domiciles, il me parait obligatoire que mon jeu de données soit capturé dans un tel environnement.

La Figure 5-2 page 83 pu être réalisée grâce au jeu de données ZBDS2023 décrit dans ce chapitre. Elle montre au cours de la journée sans attaque du 2 juillet 2022 (fuseau horaire CEST) le RSSI instantané d'un détecteur de mouvement d'identifiant 16 bits 0x0A12, capté par une des 4 sondes, celle appelée « RPI2 ». Cela représente environ 20000 trames. On remarque que pendant que la maison dort (« SOMMEIL 1 »), le RSSI reste stable autour de -80.5 dBm. Ensuite, lors du réveil (« ACTIVITÉ 1 »), celui-ci fluctue violemment sur une plage de 15 dB. Puis, lorsque les hôtes s'absentent (« ABSENCE »), le RSSI se restabilise entre -82 et -85 dBm. Consécutivement au retour des hôtes (« ACTIVITÉ 2 »), le RSSI est à nouveau très volatile sur une plage d'environ 17 dB, pour se restabiliser enfin autour de -86.5 dBm quand les hôtes se couchent (« SOMMEIL 2 »). Globalement, il est très difficile d'associer l'identifiant 0x0A12 à une valeur emblématique de RSSI ou à une distribution resserrée de celui-ci.

Il est intéressant de voir que la capture se termine avec un RSSI stabilisé mais différent de celui avec lequel elle avait commencé 24 heures plus tôt, montrant une probable réorganisation de l'environnement pendant la journée.

Ensuite, on voit que ce dispositif très simple de capture, qu'un attaquant aurait pu placer discrètement, permet d'obtenir aisément des informations liées à la vie privée. Il est en effet facile d'inférer l'activité des occupants : en journée, un RSSI stable témoigne avec une bonne probabilité d'une absence des hôtes. C'est sans doute le moment idéal pour une intrusion, physique cette fois, dans le domicile.

De façon corrélée, le tracé des distributions des RSSI par objet sur une heure la nuit et sur une heure le jour, toujours sans attaque, montre de grosses disparités statistiques du RSSI entre ces deux périodes ainsi que l'illustrent les Figure 5-3 page 84 et Figure 5-4 page 85 pour 9 objets du jeu de données ZBDS2023 décrit dans ce chapitre. Sur ces figures, on a indiqué en légende l'identifiant de chaque objet (à relier au Tableau 5-2 page 88), la valeur du RSSI qui se produit le plus et le nombre d'occurrences de celle-ci. La vie domestique de jour modifie considérablement les propriétés statistiques du RSSI (déplacement du maximum et augmentation de l'écart-type).

Ainsi, on constate que dans leur évolution temporelle, les valeurs successives de RSSI :

- ne sont pas indépendantes les unes des autres comme le seraient des tirages aléatoires ; elles font partie de séries temporelles (*time series*) présentant une certaine continuité, à exploiter en tant que telles pour prendre en compte la réalité de cet attribut ;
- ne sont pas tirées de la même distribution[21].

Ces deux caractéristiques permettent de dire que les RSSI successifs ne sont pas « indépendants et distribués identiquement ». L'acronyme anglais IID (*Independent and Identically Distributed*) est intéressant. Fort de mes observations, je peux affirmer que le RSSI d'une maison connectée vivante est une grandeur qui n'est pas IID.

Pour conclure cette section et avant de revenir au jeu de données, les fortes variations de moyenne du RSSI au cours des différentes phases de la journée ainsi que la grande volatilité de celui-ci mettent à jour des défis de taille pour construire des empreintes simples basées par exemple sur des tuples de RSSI. En outre, qu'en sera-t-il lorsque des RSSI d'objets malveillants avançant l'identifiant 0x0A12 viendront se superposer à un fond très instable ?

[21] On peut le dire d'une autre façon : dans l'évolution temporelle du RSSI, il n'y a pas de stationnarité. J'ai également pu vérifier qu'il n'y avait pas de saisonnalité.

Sera-t-il possible de les discriminer ? Les détections basées sur des seuils semblent clairement inadaptées. Des IDS subtils sont nécessaires ; disposer d'un jeu de données pour les synthétiser est d'un grand intérêt.

5.4 Caractéristiques du jeu de données ZBDS2023

Guidés par les critères énoncés dans (Cordero et al., 2021) et (Sharafaldin et al., 2018) concernant les jeux de données ayant vocation à tester, évaluer et comparer des IDS, j'ai conçu et rendu public ZBDS2023, un jeu de données compatible avec les propositions d'IDS du chapitre précédent et dont les principales caractéristiques et qualités sont listées ci-après :

1. **durée :** 10 jours. La capture a commencé le 30 juin 2022 à 17:00 (CEST) et s'est terminé le 11 juillet 2022 à 9:00 ;
2. **lieu de capture :** pour construire un jeu de données Zigbee réaliste, la capture s'est déroulée dans une maison de 100 m^2 à deux étages où les habitants vaquaient normalement à leurs activités, où les portes étaient régulièrement ouvertes et fermées, participant à un environnement physique continuellement changeant. Un four à micro-ondes était également présent et des réseau BLE et Wi-Fi coexistaient avec le réseau Zigbee. Les objets ont été laissés à leur emplacement dédié et émettant à puissance constante. Les sondes utilisées pour la capture des trames n'ont pas bougé non plus ;
3. **objets :** 10 objets Philips Hue dédiés à l'éclairage, datant de 2020 à 2022, ont été utilisés ; 4 types d'objets sur 6 sont chacun représentés par deux instances, ce qui permet d'évaluer la possibilité de distinguer des instances plutôt que des types ;
4. **sondes :** grâce à 4 sondes passives dédiées à Zigbee et distribuées dans la maison, les trames sont capturées jusqu'à 4 fois (selon la couverture et les obstacles), donnant accès à des données redondantes de couche MAC (longueurs de trame, temps inter-trames, etc.) ainsi qu'à plusieurs informations de RSSI ; le jeu de données est organisé en fichiers PCAP élémentaires d'une heure, fusionnables, triés par sonde ; avec une moyenne d'environ 1000 trames par minute pour l'ensemble des objets, le jeu de données fait environ 2.8 Gio non compressé ;
5. **attaques labellisées précisément :** c'est une qualité importante d'un jeu de données que d'avoir des situations anormales et que celles-ci soient référencées correctement afin d'établir la base de vérité (*ground truth*). Avec l'apprentissage supervisé, cela est nécessaire pour l'entraînement, la validation et le test. Avec l'apprentissage non-supervisé, cela permet de vérifier si les détections annoncées d'attaques sont effectivement correctes. Le trafic de ZBDS2023 est essentiellement bénin, permettant si besoin construire un modèle de trafic légitime pendant ces phases mais 50 attaques de 5 types différents, documentées en 5.3, ont été injectées pour évaluer des stratégies d'IDS ;
6. **évaluation de sécurité :** le 9 juillet 2022 à 15:39 (CEST), bien après le début de la capture, une deuxième télécommande (*dimmer switch*, référencée DS2 dans le Tableau 5-2 décrivant les objets) a été associée au réseau Zigbee. L'intégralité de la procédure d'association est capturée dans ZBDS2023 et permet d'obtenir la clé de réseau, donnant accès à toutes les couches hautes déchiffrées de tous les objets. Ce processus utilisant Wireshark est détaillé pratiquement en 2.4.3.2 ;
7. **métadonnées :** toutes les caractéristiques de ZBDS2023 et son élaboration sont décrites dans le présent chapitre et dans (Lourme et al., 2023) ;

(Suite après la Figure 5-4)

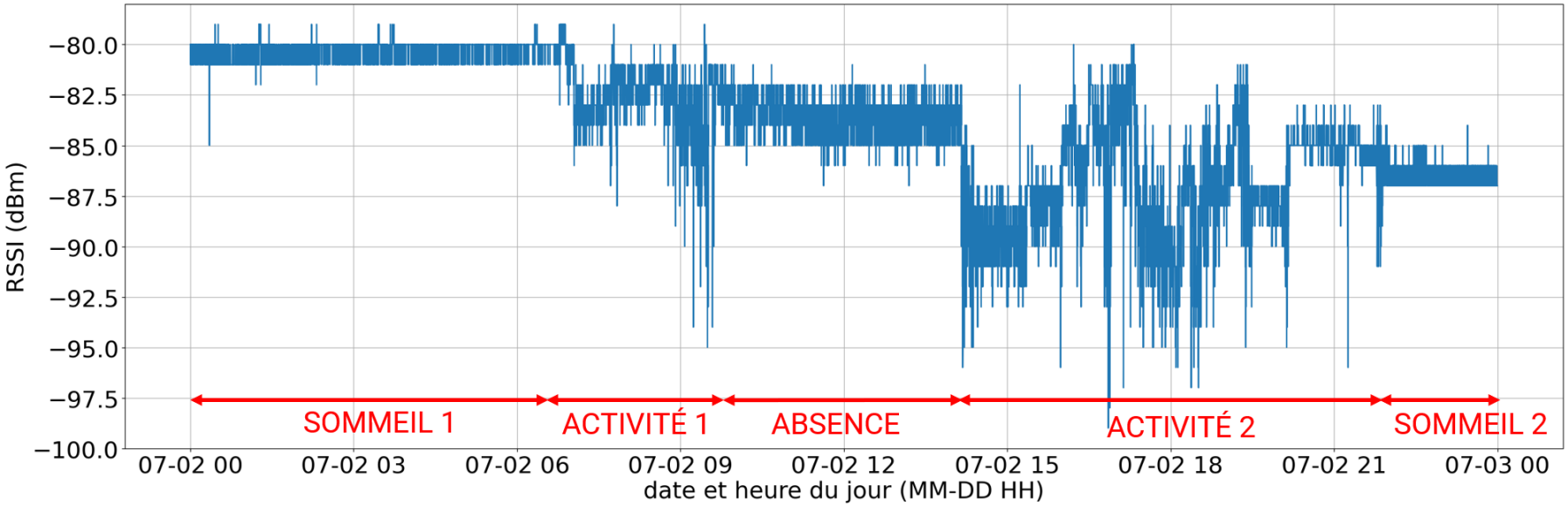

Figure 5-2. RSSI associé à l'identifiant 0x0A12, capturé par la sonde RPI2, journée sans attaque du 2 juillet 2022

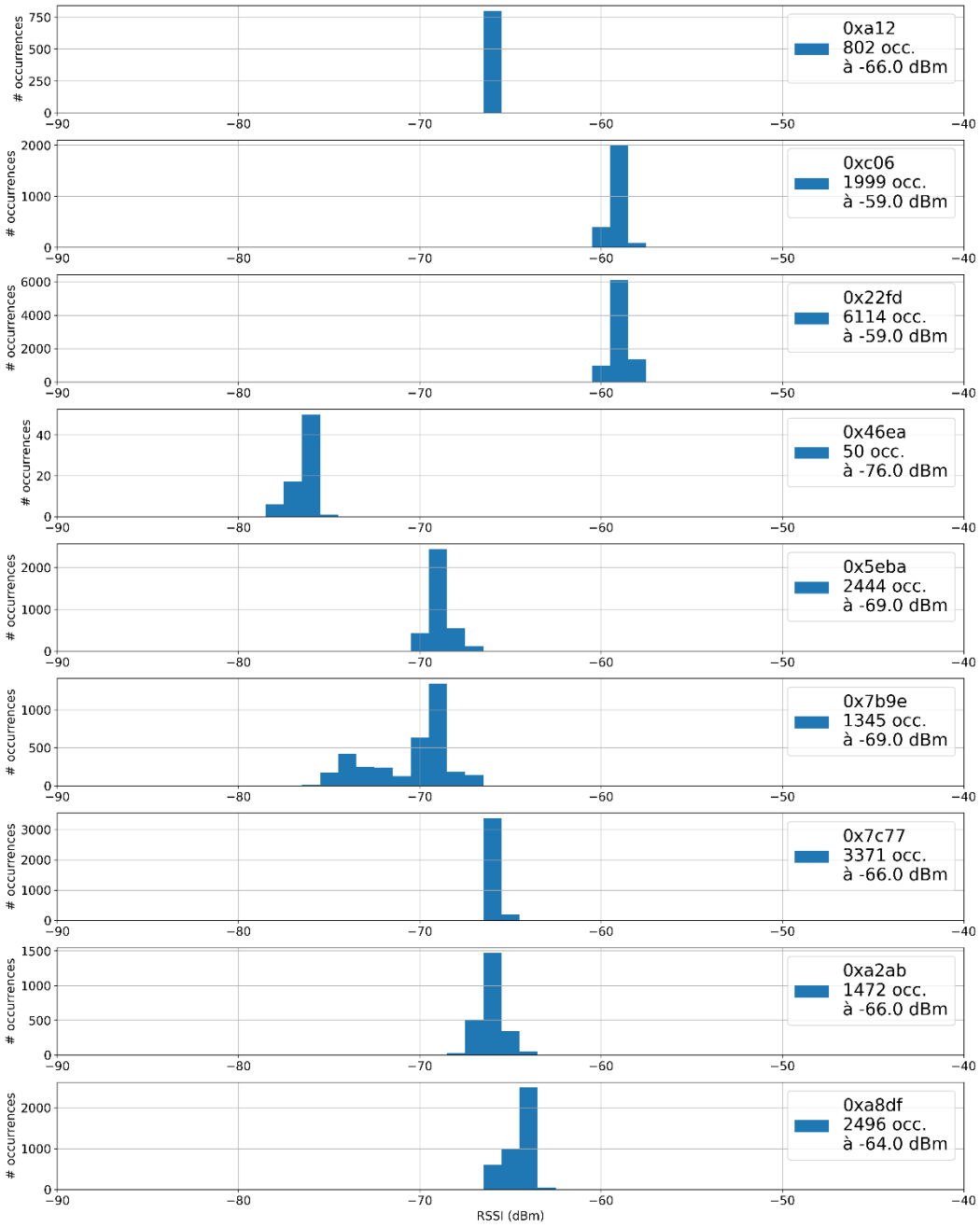

Figure 5-3. Distribution des RSSI vus de la sonde RPI3, de 02h00 à 03h00 le 1er juillet 2022, journée sans attaque

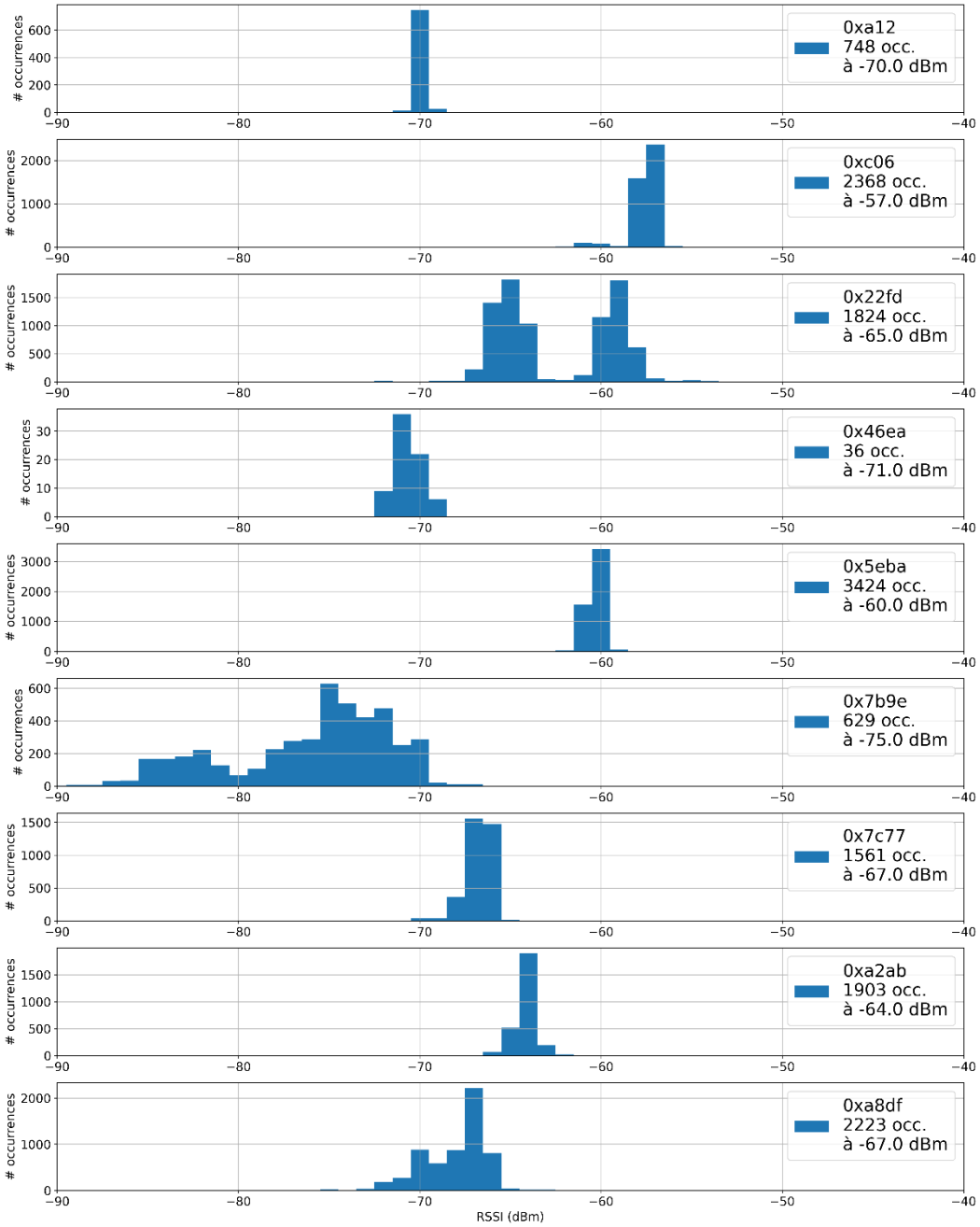

Figure 5-4. Distribution des RSSI vus de la sonde RPI3, de 16h00 à 17h00 le 1er juillet 2022, journée sans attaque

8. **disponibilité :** validé sans difficulté en janvier 2023 par le délégué à la protection des données du laboratoire CRIStAL, le jeu de données est disponible librement sur Recherche Data Gouv[22] (Lourme and Hauspie, 2023a) ; un dépôt Gitlab de ressources est également disponible[23] ; il comprend notamment la liste des attaques menées, un script Python de prétraitement et un notebook Jupyter de prise en main.

[22] https://recherche.data.gouv.fr/
[23] https://gitlab.univ-lille.fr/2xs/sdr-ids/these-olivier-lourme/zigbee-dataset

5.5 Banc d'essai

La Figure 5-5 montre l'emplacement des 4 sondes et 10 objets utilisés dans le banc d'essai permettant le jeu de données. Les sections suivantes détaillent ces deux types de ressources.

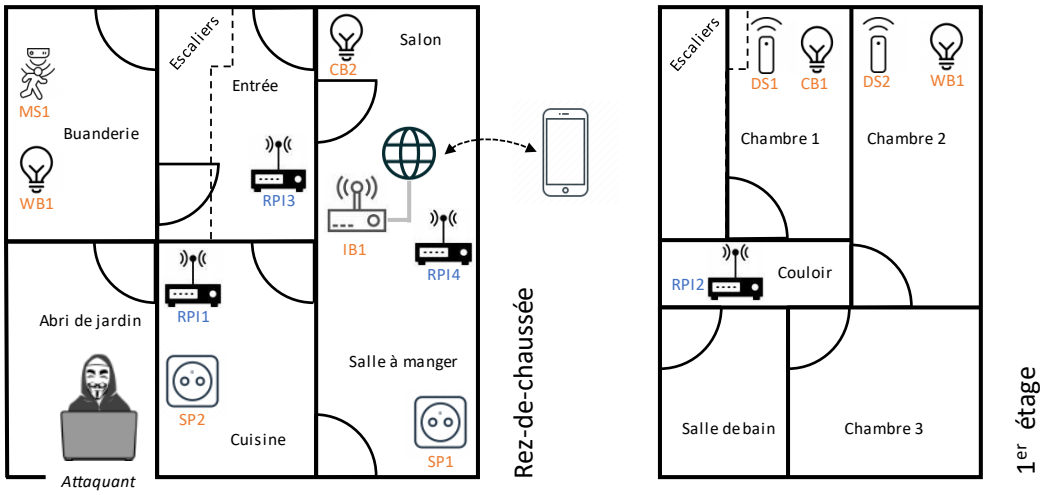

Figure 5-5. Emplacement des sondes et objets dans la maison de test

5.5.1 Sondes

Les captures sont effectuées depuis 4 emplacements différents de la maison grâce à 4 sondes référencées Figure 5-5 par les noms RPI1, RPI2, RPI3, RPI4. Chacune est construite à partir d'un émetteur-récepteur bon marché Texas Instruments CC2531 équipé d'une interface USB (CC2531 USB Dongle), connectée à un Raspberry PI 4 Model B.

Le CC2531 présente les caractéristiques données dans le Tableau 5-1 en ce qui concerne le RSSI.

Sensibilité (dBm)	Étendue de mesure (dB)	Précision (dB)	Quantification (dB)
-97	100	±4	1

Tableau 5-1. Caractéristiques de l'émetteur-récepteur CC2531 concernant le RSSI

J'ai flashé le micrologiciel `fw_cc2531.hex` mis à disposition par Texas Instruments dans chaque CC2531. Celui-ci permet, à chaque fois qu'une trame est reçue de remplacer le premier des deux octets de séquence de vérification de trame (FCS, cf. Tableau 2-2) par la valeur du RSSI en signé sur 8 bits[24]. Un décalage de -73 dB doit être additionné à cette dernière pour obtenir la valeur réelle du RSSI[25]. Pour bénéficier de cet accès aux valeurs de RSSI avec Wireshark et les outils basés dessus (e.g., la bibliothèque Pyshark), il faut

[24] Le FCS est perdu mais un bit indiquant s'il est correct est placé sur le bit de poids fort du deuxième octet de FCS.
[25] https://www.ti.com/lit/ug/swru191f/swru191f.pdf, pages 230 et 233.

préciser sous Wireshark dans ses préférences IEEE 802.15.4 que cette option de remplacement est activée, autrement tous les FCS seront indiqués faux.

Côté logiciel, *ccsniffpiper*[26] est un outil compatible avec le micrologiciel susmentionné, permettant de déclencher des captures et de les enregistrer sous forme de fichier PCAP. Pour chaque sonde, j'ai automatisé par un script *bash* le processus de lancer une nouvelle capture toutes les heures et de la nommer en accord avec la sonde et l'heure de début de la capture. Une fois décompressé, le jeu de données est constitué de 4 répertoires nommés `rpi1`, `rpi2`, `rpi3` et `rpi4`, contenant les captures d'une heure effectuées respectivement par les sondes RPI1, RPI2, RPI3 et RPI4. Chaque capture élémentaire est exploitable dans un fichier PCAP nommé par exemple ainsi :

`rpi2-2022-07-01-18-00-00.pcap`

Ce nom signifie que la capture a été effectuée depuis la sonde RPI2 et qu'elle a démarré à 18:00:00 (CEST) le 1er juillet 2022. Tous les fichiers PCAP couvrent une heure de capture, ils peuvent être fusionnés, par exemple avec Wireshark (en application graphique ou en ligne de commande), pour obtenir des captures plus longues.

Jusqu'à maintenant, il n'y a pas eu d'outil élaboré qui permette de regrouper les 4 trames relatives à un envoi mais si cela est nécessaire, cela ne devrait pas être une tâche trop complexe : avant le début de la capture, les 4 sondes ont été synchronisées en utilisant le protocole NTP et le temps *epoch* de l'arrivée de chaque trame est disponible. L'observation de ce dernier montre que l'ordre de grandeur de la différence des temps d'arrivée pour les 4 trames d'un même envoi est de seulement 10^{-4} s. Il est cela dit peu probable que l'on ait besoin de faire ce travail de regroupement. En effet, la plupart du temps, les systèmes de monitoring ou les IDS élaborent des attributs statistiques sur des fenêtres glissantes de quelques secondes, et ce sont ces grandeurs agrégées qui alimentent les modèles, classificateurs, etc.

Pour conclure cette section sur les sondes, il serait intéressant de voir laquelle d'entre elles a capturé le plus de trames et de noter les caractéristiques de son emplacement pour le cas où on souhaite construire un IDS bon marché à une seule sonde. Sur mon jeu de données, il s'agit de la sonde RPI2. Il semblerait que sa position centrale en haut de l'escalier lui permette une bonne couverture. C'est une information intéressante pour l'utilisateur car, rappelons-nous, on ne souhaite pas qu'un expert vienne installer l'IDS au domicile.

5.5.2 Objets

5.5.2.1 Caractéristiques des objets

Les caractéristiques des objets visibles sur la Figure 5-5 sont données dans le Tableau 5-2. L'identifiant court sur 16 bits est celui qui a été obtenu après une procédure d'association IEEE 802.15.4. Tous les objets avaient déjà un tel identifiant quand la capture a commencé, à l'exception de DS2 qui a été associé plus tard pour capturer son processus d'association (afin d'obtenir éventuellement la clé de réseau). La distinction RFD/FFD a été présentée en 2.4.2.2.

Précisons également que l'identifiant court du réseau (PAN ID) est `0xB7C5` et l'identifiant étendu du réseau (Extended PAN ID) est `47:4D:CE:61:BF:05:E6:EF`. Le canal utilisé est le n°20.

[26] https://github.com/andrewdodd/ccsniffpiper

5.5.2.2 « Liens » entre objets

Les relations suivantes de « liens » (*bindings*) entre objets ont été établis avec l'application Philips Hue pour smartphone. Ces « liens » provoquent des interactions entre les objets, participant ainsi à un jeu de données réaliste :

- WB1 s'allume quand MS1 détecte un mouvement ;
- CB1 est contrôlé par la voix depuis un Google Home (ce dernier communique avec la passerelle IB1 en Wi-Fi, puis IB1 contrôle CB1) ;
- DS1 contrôle CB1 ;
- DS2 contrôle WB2 (après que DS2 ait rejoint le réseau).

En outre, tout au long de la capture, l'application Philips Hue pour smartphone a été utilisé intensément pour contrôler les objets via IB1, depuis l'intérieur ou l'extérieur de la maison.

Réf.	Id. 16 bits	Type (anglais)	Type (français)	Adresse MAC	RFD/FFD
MS1	0x0A12	Motion Sensor	Capteur de mouvement	00:17:88:01:0B:CD:0C:32	RFD
SP2	0x0C06	Smart Plug	Prise	00:17:88:01:08:D8:9B:69	FFD
IB1	0x22FD	Internet Bridge	Pont Internet	00:17:88:01:05:11:BD:4A	FFD
DS1	0x46EA	Dimmer Switch	Télécommande	00:17:88:01:08:F0:4C:D7	RFD
CB2	0x5EBA	Color Bulb	Lampe couleur	00:17:88:01:06:A1:66:0B	FFD
WB2	0x7B9E	White Bulb	Lampe blanche	00:17:88:01:08:B6:20:FE	FFD
WB1	0x7C77	White Bulb	Lampe blanche	00:17:88:01:08:BF:D3:B7	FFD
DS2	0x88C2	Dimmer Switch	Télécommande	00:17:88:01:0B:DA:40:CE	RFD
SP1	0xA2AB	Smart Plug	Prise	00:17:88:01:08:D8:B9:B8	FFD
CB1	0xA8DF	Color Bulb	Lampe couleur	00:17:88:01:06:A1:5A:00	FFD

Tableau 5-2. Liste des objets Zigbee présents dans le jeu de données ZBDS2023

5.6 Modèle de menace, attaques injectées

Pour favoriser le développement d'IDS dédiés à Zigbee et notamment des systèmes de détection d'usurpation d'identité, le jeu de données proposé inclut différents types d'attaques de communication. Celles-ci ont été choisies comme représentatives de ce qu'un attaquant peut entreprendre avec peu d'expertise et de ressources. Une bonne partie des attaques se déroulent dans un mode inondation. L'épuisement des ressources résultant et le manque de réactivité dont font preuve alors les objets et le réseau peuvent être un préjudice suffisant en lui-même ou être une étape dans une séquence d'attaques, comme celle conduisant à l'obtention de la clé de réseau décrite en 2.4.3.2 ou comme d'autres décrites

dans (Sadikin et al., 2020). Comme j'ai utilisé des objets Zigbee hauts de gamme, toutes les attaques ne sont pas couronnées de succès (par exemple celles de rejeu, déjouées par un compteur de trames) mais la détection de ces tentatives d'intrusions est néanmoins un point crucial ; l'intérêt d'un jeu de données est de permettre l'élaboration des IDS en relation.

Les conditions suivantes ont été observées :

- Comme sur la Figure 2-4 page 36 ou la Figure 5-5 page 86, l'attaquant est dans le voisinage immédiat du réseau attaqué. Comme les objets et les sondes, il reste à une position fixe. Il est capable de prendre l'identité souhaitée en forgeant des trames malicieuses avec l'identifiant 16 bits d'un objet légitime. Ces trames sont émises à puissance constante en utilisant le cadriciel offensif *killerbee*[27] depuis un ordinateur sous Ubuntu. L'émetteur-récepteur utilisé est un autre CC2531 avec interface USB mais flashé cette fois avec le micrologiciel offensif *bumblebee*[28] ; sa puissance maximum d'émission est de 2.8 mW (4.5 dBm). Il n'a pas été possible d'utiliser le cadriciel offensif Mirage (Cayre et al., 2019) car concernant Zigbee, celui-ci utilise un émetteur-récepteur RZ RAVEN USB stick qui n'est plus disponible.
- L'attaquant n'a pas de connaissance de la clé de réseau. Il peut seulement écouter le réseau pour acquérir de la connaissance à son sujet (objets, identifiants, etc.) ou injecter des messages pour pratiquer des attaques-inondations de couche MAC ou des attaques de rejeu.

Pendant les 10 jours de capture, jai initié sur 6 d'entre eux 10 sessions de 5 types d'attaques. Chaque session est numérotée entre 1 et 10 et chacune des 5 types d'attaques est étiquetée par une lettre de A à E (les types A et B sont néanmoins identiques, il faut les faire se succéder pour obtenir une attaque d'une minute, durée commune aux autres types d'attaque). Réaliser 10 fois les mêmes attaques au cours de l'intégralité de la capture est justifié car l'environnement physique de la maison change souvent. Le Tableau 5-3 donne une description des 5 types d'attaques conduites au cours de chaque session.

Les attaques sont caractérisées par leur numéro de session, leur étiquette de type et leur date de début (leur durée est indiquée dans le Tableau 5-3). En Python, avec la librairie *pandas*, le dictionnaire suivant définit par exemple l'attaque de type E qui a eu lieu pendant la session n°9 :

```
{
  'sess': '09',
  'type': 'E',
  'start': pandas.Timestamp('2022-07-08 20:25:10+02:00')
}
```

La liste complète des attaques présentées de cette manière est disponible depuis le fichier `attacks_references.py` du dépôt Gitlab de prise en main (cf. fin de 5.4).

[27] https://github.com/riverloopsec/killerbee
[28] https://github.com/virtualabs/cc2531-killerbee-fw

Type	Descriptif	Détails de l'attaque	Commentaires
A et **B**	Inondation de requêtes d'Association (*Association Request Flood*)	En émettant une trame *Beacon Request*, l'attaquant demande à tous les routeurs de l'informer sur les caractéristiques du réseau auquel chacun appartient. Le premier à répondre, en général la passerelle Internet, va gérer le processus d'association. À cette occasion, l'attaquant adopte une adresse MAC 64 bits aléatoire et reçoit finalement un identifiant 16 bit. Une répétition de cette demande d'association en mode inondation est une attaque d'usurpation d'identité de type **sybil** qui consomme l'ensemble des identifiants 16 bits disponibles pour les RFD et FFD.	Durée ≈ 30 s. Entre 35 et 50 requêtes. Un message *Permit Join* de validité 30 secondes a été préalablement envoyé à tous les routeurs par l'application du smartphone.
C	Attaque de rejeu (*Replay Attack*)	Dans cette attaque de type **mascarade**, l'attaquant usurpe la passerelle Internet IB1 pour prendre le contrôle de l'objet CB2. Un fichier PCAP élémentaire a capturé pendant 10 secondes plusieurs ordres de la part d'IB1 pour allumer et éteindre l'ampoule couleur CB2, au milieu des autres échanges normaux. Ce fichier élémentaire est rejoué 6 fois de suite par l'attaquant pour constituer l'attaque (commande `zbreplay` de *killerbee*). Dans le fichier PCAP élémentaire, des trames ayant un autre propos qu'allumer et éteindre CB2 ont aussi été capturées. Par exemple, le capteur de mouvement MS1 effectue 2 *Data Requests* normaux à son parent WB1, ce qui fait 12 *Data Requests* également issus de l'attaquant pendant cette attaque. Cet effet « secondaire » produit une sorte d'attaque de type E à faible fréquence.	Durée ≈ 1 min. Les ordres d'allumage et d'extinction émis par IB1vers CB2 proviennent de l'application du smartphone.
D	Inondation de requêtes de données (*Data Request Flood*) 1er type	Dans cette attaque d'usurpation de type **mascarade**, l'attaquant prend l'identifiant de la télécommande DS1 pour forger des centaines de Data Requests visant l'ampoule couleur CB1. Cette dernière n'est pas le parent de DS1, il va donc lui envoyer des trames *Leave* l'invitant à quitter le réseau. Si DS1 est configuré pour traiter ces trames *Leave*, c'est un moyen pour l'attaquant de faire quitter le réseau à DS1.	Durée ≈ 1 min. Environ 500 *Data Requests*.
E	Inondation de requêtes de données (*Data Request Flood*) 2ème type	Dans cette attaque d'usurpation de type **mascarade**, l'attaquant prend l'identifiant du capteur de mouvement MS1 pour forger en son nom des centaines de trames Data Request visant l'ampoule blanche WB1, le parent légitime de MS1.	Durée ≈ 1 min. Environ 500 *Data Requests*.

Tableau 5-3. Types d'attaques conduites au sein d'une session et leur description

5.7 Cas d'étude : IDS naïf pour la détection d'usurpation

Plusieurs cas d'utilisation du jeu de données sur des périodes sans attaque ont déjà été présentés jusqu'ici. Effectivement, j'ai pu :

- faire du partitionnement d'empreintes d'objets et montrer le principe de la détection des usurpations d'identité (cf. 4.3.2.2) ;
- exposer l'évolution temporelle du RSSI d'un objet capté par une sonde au cours d'une journée complète (cf. 5.3.4) ;
- comparer des distributions de RSSI entre le jour et la nuit (cf. 5.3.4).

Dans cette section, pour prendre en main de façon complète l'exploitation du jeu de données en mode hors-ligne, je propose un IDS naïf basé sur des seuils de RSSI appliqué à des RSSI glissants, afin de détecter les attaques d'usurpation d'identité.

5.7.1 Attaques retenues

Le 8 juillet 2022, les attaques de mascarade 8C, 9C, 10C, 8E, 9E et 10 E sont injectées sur le réseau en plus des activités normales. Pendant ces attaques, l'attaquant utilise l'identifiant du capteur de mouvement MS1, enfant légitime de l'ampoule blanche WB1, pour envoyer des requêtes de données illégitimes à WB1, le parent de MS1. Les deux types d'attaques C et E durent une minute mais les attaques C sont à faible fréquence (12 trames par minute) alors que les attaques E sont à haute fréquence (500 trames par minute) ; on peut se référer au Tableau 5-3 pour plus de détails.

5.7.2 Préparation des données

Pour préparer les données, je procède comme suit :

1. **Création d'un fichier de captures fusionnées –** Je fusionne les 24 fichiers PCAP de la journée considérée (8 juillet 2022), capturés par la sonde RPI2 uniquement, afin d'obtenir un fichier PCAP de 24 heures contenant 1 378 154 trames.
2. **Filtrage des trames** – Puis, par un script en langage Python nommé `zigbee_dataset_preprocess.py`, basé sur la bibliothèque *pyshark* et accessible depuis le dépôt Gitlab de prise en main (cf. 5.4), je prétraite le fichier PCAP afin d'exclure les trames malformées (environ 0.1% du nombre total de trames), celles avec une mauvaise séquence de vérification de trame FCS (environ 1.2%), celles qui n'ont pas d'identifiant source 16 bits (environ 41.4%) par exemple les nombreuses trames d'acquittement ou les rares trames *Beacon Request*) et enfin les quelques trames contenant des valeurs aberrantes de RSSI. 787 481 trames sont finalement retenues (environ 57.1%). Pour chacune, je ne conserve qu'une liste de 6 champs ou métadonnées, renseignés dans le Tableau 5-4. Le fichier généré (au format `json`) contient finalement une liste de listes qui servira de données d'entrée à l'IDS.

Champ	`"packet_idx"`	`"src16"`	`"length"`	`"time_rel_us"`	`"rssi"`	`"epoch_us"`
Propos	numéro de la trame, commence à 0	identifiant source sur 16 bits	longueur de la trame (octets)	temps passé depuis la première trame (µs)	valeur du RSSI (dBm)	temps *epoch* (µs)

Tableau 5-4. Champs conservés pour les trames retenues lors du prétraitement

5.7.3 Conception de l'IDS

Les étapes de travail suivantes détaillent la méthodologie retenue pour l'IDS :

3. **Récupération des trames associée à l'identifiant considéré –** Dans un notebook Python basé sur des datagrammes *pandas* dont les champs sont donnés Tableau 5-4, l'IDS est implémenté : je retiens uniquement les 20 266 trames associées à l'identifiant 16 bits de MS1, c'est à dire 0x0A12.

4. **Calcul d'une moyenne glissante –** Comme montré sur la Figure 5-6 en bleu pale, le RSSI instantané est très volatile et apparaît inapproprié pour des modèles très simples basés sur des dépassements de seuils. Pour adresser cela tout en prenant en compte la dimension temporelle des séries de RSSI, je calcule la moyenne glissante du RSSI sur des fenêtres de 30 éléments, un nombre assez grand pour adoucir celui-ci mais assez petit pour rester sensible aux changements abrupts. Le graphe résultant est reporté sur la Figure 5-6 en bleu foncé.

5. **Calcul d'un RSSI de référence –** Pour établir une valeur de référence pour le RSSI de MS1, j'exploite la relative stabilité présentée par le RSSI pendant les heures de nuit où l'environnement physique est statique. Le calcul de sa moyenne entre minuit et 6 heures du matin donne -86 dBm.

6. **Fixation des seuils de détection autour du RSSI de référence et pratique de la détection –** Ensuite, il faut choisir une paire de seuils autour de la moyenne de nuit du RSSI calculée précédemment. Concernant la détection d'attaque, si la moyenne glissante du RSSI sort de la bande limitée par les seuils, je considère qu'il y a une anomalie, c'est-à-dire une attaque d'usurpation. La largeur de la bande a été finalement choisie à ±4 dB car cela donne les meilleures métriques de détection. C'est bien sûr une approche de conception non réaliste mais le but ici est d'établir une référence à laquelle des algorithmes plus subtils pourront se comparer.

5.7.4 Résultats

Les résultats sont obtenus par cette dernière étape :

7. **Calcul des métriques –** Pour élaborer les métriques de détection, je découpe les 24 heures en tranches de 10 minutes pour corréler au sein de chacune d'elles les attaques effectives et les anomalies détectées, afin d'établir les nombres de, respectivement, vrais négatifs, vrais positifs, faux négatifs et faux positifs, notés respectivement *TN*, *TP*, *FN* et *FP*, puis les métriques classiques définies en 2.5.3 comme la justesse, la précision et le rappel. Notons qu'il n'y a jamais plus d'une attaque effective par tranche et qu'une seule anomalie suffit à évaluer la tranche comme attaquée. Sur la Figure 5-6, exceptées les occurrences de vrais négatifs, très nombreuses, celles de vrais positifs, faux négatifs et faux positifs sont inscrits en gras noir par **TP**, **FN** et **FP**.

Présentés dans le Tableau 5-5 page 94, les résultats sont intéressants en dépit de l'extrême simplicité et du caractère bon marché du détecteur. En effet, 5 attaques d'usurpation ont été détectées sur les 6 réalisées conduisant à un rappel honorable de 83.3%. Bien-sûr, il reste une marge de progression en ce qui concerne les trop nombreux faux positifs mais cette première synthèse d'IDS me donne envie d'aller plus loin dans l'exploitation du RSSI pour la détection d'intrusion.

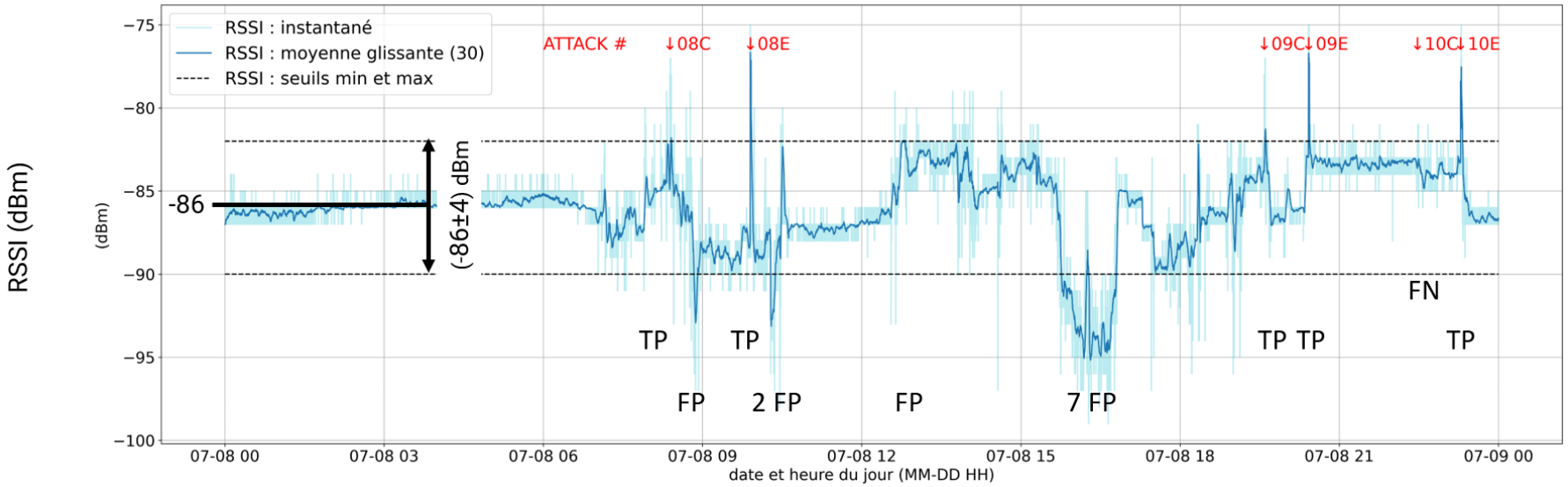

Figure 5-6. RSSI instantané et glissant associé à l'id. 0x0A12, capturé par la sonde RPI2, journée avec 6 attaques du 8 juillet 2022

On saisit cela dit les limites de l'utilisation d'une seule sonde. Dans un tel contexte, si par malchance le RSSI « emblématique » de l'attaquant est proche de celui de l'objet usurpé, même momentanément à cause d'un changement d'environnement passager, des intrusions ne sont pas détectées. Cet échappement de la détection est à l'origine de faux négatifs (ici un seul, lors de l'attaque 10C). Utiliser au moins une sonde en plus permettrait de réduire cet inconvénient.

TN	***TP***	***FN***	***FP***	***Justesse***	***Précision***	***Rappel***	***TNR***	***FPR***
127	5	1	11	91.7%	31.2%	83.3%	92.0%	8.0%

Tableau 5-5. Métriques de détection de l'IDS basé sur des seuils et des moyennes glissantes de RSSI

5.8 Conclusion

Ce chapitre a d'abord exposé l'intérêt de disposer lors de la synthèse d'IDS de jeux de données de qualité représentatifs des contextes où ces systèmes vont s'insérer, aussi bien en termes d'environnement physique que d'attaques susceptibles de survenir. En outre, les attaques reproduites dans ces jeux de données doivent être correctement labellisées afin de pouvoir quantifier par des métriques l'efficacité des IDS réalisés et ainsi faciliter leur comparaison.

Rappelant les motivations qui m'ont poussé à développer un jeu de données pour le protocole Zigbee dédié aux maisons connectées, j'ai ensuite présenté progressivement les caractéristiques du RSSI, attribut physique original du jeu de données, pouvant participer à la résolution des problématiques d'identification et d'usurpation d'identité, surtout s'il est mesuré de plusieurs endroits. J'ai également montré que le RSSI, certes facile d'accès, était une grandeur très volatile pendant les phases d'activité de la maison, rendant impossible la synthèse d'empreintes d'objet basées sur des RSSI stables, ceux-ci ne l'étant pas.

Ainsi, ces différentes constatations m'ont donné les lignes directrices de la conception d'un jeu de données réaliste nommé ZBDS2023 qui, je l'espère, sera utile à la communauté. Une dizaine d'objets sont présents et toutes les trames échangées sont capturées depuis 4 sondes distribuées. Des attaques représentatives de ce qu'il est possible de faire pour un attaquant basique ont été injectées. Ainsi, des propositions d'IDS ou de systèmes de monitoring disponibles dans la littérature utilisant des attributs de couche MAC sont réalisables pour le protocole Zigbee grâce à mon jeu de données, proposé en accès libre. 4 RSSI capturés depuis les 4 sondes sont aussi disponibles. Des IDS innovants peuvent être envisagés.

Enfin, un cas d'IDS simple a été proposé pour illustrer du début à la fin l'usage du jeu de données. Bien que prometteur, cet exemple a rappelé qu'en raison de sa volatilité en environnement changeant, le RSSI était un attribut non trivial à exploiter dans le cadre de la détection d'usurpation. Le chapitre suivant s'intéresse justement à des algorithmes pouvant tirer parti du RSSI malgré cette caractéristique.

Chapitre 6 - Détection des usurpations d'identité par réseaux de neurones récurrents

6.1 Introduction : Approche de la détection

Dans les problématiques d'authentification et de détection d'usurpation, il est nécessaire de travailler avec des attributs de couche physique. L'intérêt du RSSI est d'être un de ces attributs qui, contrairement aux autres du même type, est facile d'accès et bon marché. Malheureusement, le chapitre précédent nous a montré que pendant les phases d'activité de la maison, où l'environnement change beaucoup, le RSSI devenait une grandeur très volatile avec laquelle il est impossible d'établir des empreintes d'objets simples, exploitables pour servir dans la détection d'intrusion.

Nous savons maintenant que dans ces environnements dynamiques, les valeurs successives de RSSI s'inscrivent dans des séries temporelles et sont issues de distributions non identiques. Plutôt que de caractériser les objets par des empreintes simples, il devient nécessaire de mener une approche basée sur des modèles des séquences temporelles normales de RSSI. Par « normales », j'entends des séquences pendant lesquelles il n'y a pas d'anomalie, c'est-à-dire pas d'attaque. En test, l'IDS différentiera les séquences bénignes de RSSI, qui obéissent au modèle, des séquences malicieuses pour lesquelles il lèvera une alarme.

Lors du Chapitre 4, l'IDS comportemental à apprentissage non-supervisé dont l'entraînement se fait sur des données d'une seule classe est apparu comme le plus adapté aux exigences recherchées de coût et de praticité. Un algorithme comme *One-class Support Vector Machine* qui se range dans cette catégorie produit cela dit un résultat décevant, ne s'appuyant que sur les valeurs isolées de RSSI sans prendre en compte le lien temporel entre celles-ci (Madani and Vlajic, 2021). En outre, il n'est pas adapté au traitement de grandes quantités de données et converge lentement. Pour s'affranchir de ces deux dernières contraintes, le choix de l'apprentissage profond (*deep learning*) permis par les réseaux de neurones s'impose, certes au prix de complexités temporelle et spatiale supérieures.

Il est finalement nécessaire 1) de retenir un algorithme non supervisé à base de réseau de neurones effectuant son entraînement sur des données d'une seule classe et 2) d'opter pour un algorithme qui sache travailler avec des séries temporelles et non pas des points indépendants les uns des autres.

La première contrainte m'encourage à utiliser un réseau de neurones de type auto-encodeur et la deuxième me pousse à considérer les réseaux de neurones récurrents (*Recurrent Neural Networks*, RNN) ou une de leurs améliorations, par exemple les réseaux à cellules LSTM (*Long Short-Term Memory*).

Au niveau du plan suivi par ce chapitre, je commence par un arrière-plan théorique 1) sur l'auto-encodeur et 2) sur les neurones récurrents et les cellules LSTM. À la fin de celui-ci, je montre l'intégration de ces cellules dans un auto-encodeur dans le but de générer des modèles de séquences normales. J'aborde ensuite la conception intégrale d'un IDS capable de détecter pour chaque objet les séquences anormales de RSSI, symptômes d'une attaque d'usurpation d'identité. Ses métriques sont évaluées puis des pistes pour les améliorer sont présentées.

6.2 Auto-encodeur, réseaux de neurones récurrents

La présente section fournit en 6.2.1 l'arrière-plan théorique concernant les auto-encodeurs (travaillant sur des données indépendantes) dans la détection d'anomalies puis en 6.2.2 celui concernant les réseaux de neurones récurrents (travaillant sur des données de type série temporelle). Je m'appuie entre autres sur les ouvrages (Géron, 2019) et (Chollet, 2021).

Notations

Dans la suite de ce manuscrit, un scalaire est noté par une lettre minuscule, par exemple x_0 ou m. Il en est de même pour une fonction, par exemple f. Un vecteur est noté par une lettre majuscule, par exemple X.

La valeur d'un scalaire, d'un vecteur ou d'une matrice à l'étape temporelle t est noté avec la notation [t]. Par exemple, $X[t]$ est la valeur du vecteur X à l'étape temporelle t. $X[t\text{-}1]$ est sa valeur passée, c'est-à-dire sa valeur à l'étape temporelle t-1.

6.2.1 Principe de l'auto-encodeur pour la détection d'anomalies

6.2.1.1 Neurone artificiel, réseau de neurones artificiels

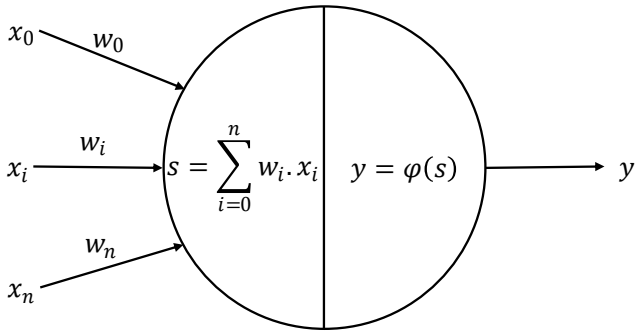

Figure 6-1. Un neurone artificiel

Composant de base des réseaux de neurones servant pour l'apprentissage profond, le neurone artificiel de la Figure 6-1 s'inspire du perceptron décrit par le psychologue américain Frank Rosenblatt en 1958. Chacune des n entrées x_1 à x_n est affectée d'un poids w_1 à w_n ; le cas de x_0 de valeur unitaire est particulier, sa multiplication par w_0 représente le « biais » (terme constant). La somme pondérée s est calculée pour ensuite alimenter une fonction φ non linéaire d'activation choisie parmi *Heaviside*, *ReLu*, *Sigmoid* (i.e., *logistic*), ou *tanh* (pour les plus connues) dont la sortie correspond à la sortie du neurone (Rosay, 2022). La constitution de couches de tels neurones puis l'empilement de celles-ci permettent de réaliser les réseaux de neurones artificiels (*Artificial Neuron Network*, ANN) que nous connaissons aujourd'hui, notamment les auto-encodeurs.

Une notation vectorielle permet d'écrire de façon compacte le travail effectué par le neurone :

Si X est le vecteur des entrées du neurone :

$$X = \begin{bmatrix} x_0 \\ \cdots \\ x_i \\ \cdots \\ x_n \end{bmatrix} \quad (6\text{-}1)$$

et si *W* est le vecteur des poids liés aux entrées :

$$W = \begin{bmatrix} w_0 \\ \cdots \\ w_i \\ \cdots \\ w_n \end{bmatrix} \tag{6-2}$$

Alors la sortie *y* du neurone s'exprime par :

$$y = \varphi(X^T W) \tag{6-3}$$

6.2.1.2 Auto-encodeur

Les auto-encodeurs, dont un exemple simple est donné Figure 6-2, sont des réseaux de neurones artificiels capables d'apprendre des représentations efficaces des données d'entrée, appelées **codages**, sans aucune supervision. En effet, le réseau est entraîné pour reconstruire à partir de la représentation des entrées des sorties les plus proches possible des entrées. Ces codages ont le plus souvent une dimension plus faible que celle des données d'entrées, c'est-à-dire que chaque instance du jeu de données a une représentation avec moins d'attributs que son entrée mais nous verrons que ce n'est pas la seule option.

Considérons un jeu de données constitué d'un nombre *m* d'instances (typiquement quelques dizaines de milliers d'unités) où chacune d'entre elles comporte *n* attributs (c'est-à-dire qu'une instance est un vecteur de dimension *n*). Par exemple, on aura *n* = 6 si chaque instance comporte 3 traitements statistiques différents sur des longueurs de trames et 3 traitements statistiques différents sur des temps inter-trames, chacun de ces traitements étant effectué sur une fenêtre temporelle de quelques secondes. L'auto-encodeur de la Figure 6-2 fait figurer *n* = 6 entrées et une couche de sortie avec également *n* neurones ; cette égalité des dimensions de l'entrée et de la sortie est une caractéristique de l'auto-encodeur.

La couche n°1 est celle des entrées. Par abus de langage, on dit que c'est celle des « neurones d'entrée » mais ce ne sont que des sources de données. La couche n°5 est celle des neurones de sortie. Les couches n°2, n°3 et n°4 sont les couches internes ou cachées. Les couches n°1 et n°2 constituent l'**encodeur**. Au centre, couche n°3, se trouve l'« espace latent » où se situe le codage (de dimension 2 pour notre exemple) de l'instance traitée. Les couches n°4 et n°5 constituent le **décodeur**. L'auto-encodeur décrit est par conséquent l'empilement d'un encodeur et d'un décodeur. En outre, sur l'exemple de la Figure 6-2, les couches sont « pleinement connectées » ou « denses », car tous les neurones ou entrées d'une couche sont connectés aux neurones de la couche suivante.

Le terme « profond » dans l'expression « apprentissage profond » (*deep learning*) est relatif à la succession des couches de neurones dans le réseau.

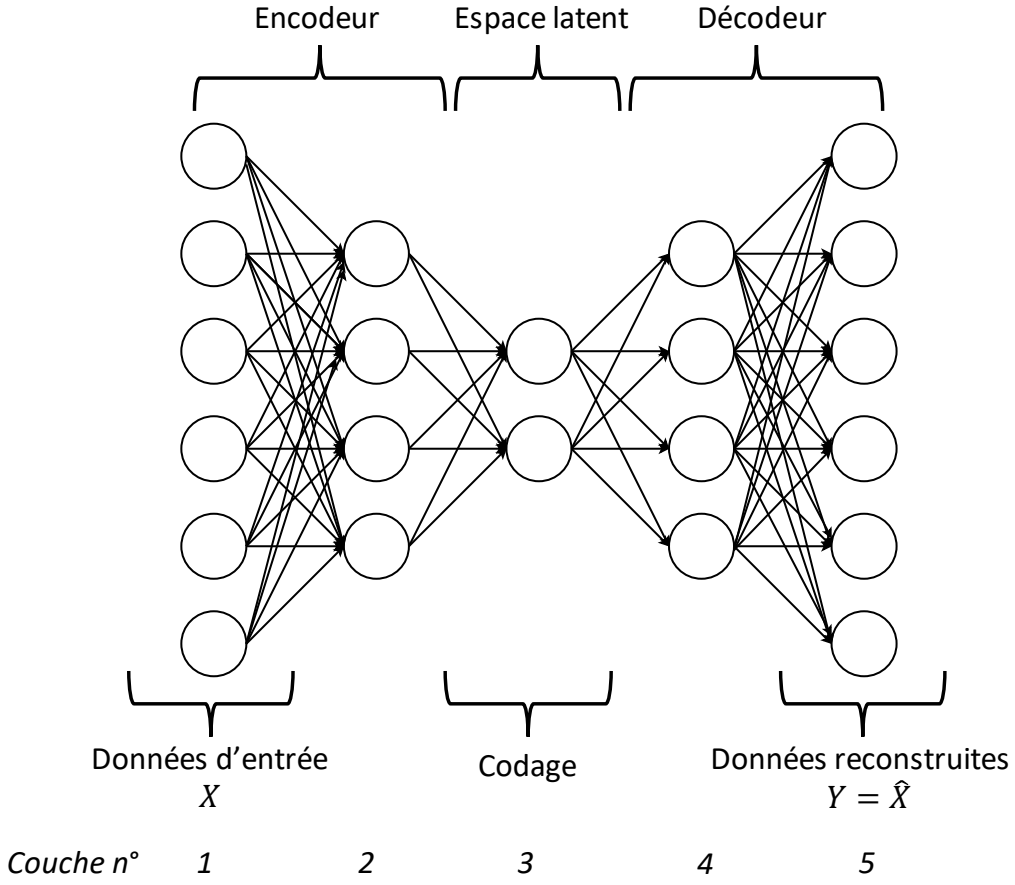

Figure 6-2. Auto-encodeur à base de neurones artificiels

6.2.1.3 Entraînement (ou apprentissage)

Dans le cadre de la détection d'anomalies, l'auto-encodeur est entraîné sur un jeu de données d'entraînement qui ne traduit que des situations bénignes, c'est-à-dire sans attaque. Plusieurs itérations (*epochs*) sur ce jeu d'entraînement mélangé sont réalisées afin d'ajuster au mieux les poids des neurones de l'auto-encodeur, ceci dans le but de minimiser l'**erreur de reconstruction** entre les sorties apprises et les entrées. Cette erreur, appelée plus généralement **fonction de coût** ou **fonction de perte**, est évaluée sur des lots dont la taille (*batch size*) est typiquement de quelques dizaines. Généralement, c'est l'erreur quadratique moyenne (*Mean Squared Error*, MSE) ou bien l'erreur absolue moyenne (*Mean Absolute Error*, MAE) qui sont calculées sur chaque lot pour faire progresser le système vers des poids optimum pour lesquelles les sorties sont les plus proches possibles des entrées. L'heuristique utilisée est une descente de gradient ou une de ses optimisations, par exemple Adam. Lors de l'entraînement des réseaux profonds, les phénomènes d'explosion / disparition des gradients sont courants et des méthodes pour les maîtriser doivent être employées (initialisation particulière des poids, fonctions d'activation spécifiques, normalisation des lots).

Éventuellement, une petite fraction (10 à 20%) du jeu de données bénin peut être réservée pour constituer un jeu de données de validation, sur lequel il n'y a pas d'entraînement mais qui sert à superviser l'évolution de l'erreur au fil des *epochs*.

6.2.1.4 Détection d'anomalies (ou test)

Dans le cadre de la détection d'anomalies, l'assomption classique quand on utilise un auto-encodeur est de dire que si une sortie est très différente de l'entrée correspondante (i.e., si l'erreur de reconstruction dépasse un certain seuil), cela signifie que l'entrée est une anomalie. En effet, si le système n'a pas réussi à reconstruire convenablement l'entrée, c'est qu'il ne l'a pas intégrée lors de son apprentissage des situations bénignes. Il est nécessaire que cet apprentissage ait donc lieu sur des situations représentatives de vie « normale ». Pour évaluer l'efficacité de la détection, le jeu de données de test doit être généré spécialement. En effet, à la différence des jeux d'entraînement et de validation, il comprend aussi des attaques ; celles-ci doivent être référencées correctement pour calculer les métriques de détection.

6.2.1.5 Autres topologies d'auto-encodeurs et autres contraintes

Il peut sembler trivial de demander à l'auto-encodeur de « recopier » ses entrées sur ses sorties. En fait, dans l'exemple de la Figure 6-2 qui nous a servi de support jusqu'à maintenant, la dimension plus petite de l'espace latent par rapport à celle de l'entrée, conduisant à un auto-encodeur dit « sous-complet », est une contrainte importante qui fait que l'auto-encodeur est obligé de trouver des représentations pertinentes des entrées[29]. Cela dit, l'auto-encodeur ne doit pas être excessivement sous-complet sous peine d'échouer à reconstruire les entrées.

À l'opposé, on trouve des auto-encodeurs « sur-complet » où la dimension de la couche de codage est supérieure à celles des entrées et sorties. On peut même se passer de couche de codage. En fait, à condition de respecter un nombre de sorties égal à celui des entrées, toutes les topologies de couches et de neurones par couche sont envisageables a priori, l'important étant d'éviter que l'auto-encodeur recopie ses entrées d'entraînement sur ses sorties sans apprendre aucune représentation des données. Cela conduirait à un surajustement des données d'entraînement avec une incapacité à généraliser sur des données de validation ou de test. Pour cela, on peut lui fixer d'autres types de contraintes, ou « régularisations », que celle utilisée jusqu'à maintenant. On trouve par exemple :

- la régularisation par *dropout* : pendant l'entraînement, il est possible de rendre certains neurones éteints (*dropped out*) sur chacune des couches (y compris sur celle d'entrée, mais pas sur celle de sortie). On fixe pour chacune d'elle un taux d'extinction, typiquement entre 20% et 50% ;
- la régularisation par ajout d'un bruit gaussien sur les entrées : l'auto-encodeur « débruiteur » doit apprendre à reconstruire proprement les entrées malgré la présence de bruit.

Comme les *dropouts*, l'arrêt précoce (*early stopping*) est une méthode de régularisation générale aux réseaux de neurones, applicable aux auto-encodeurs : pendant l'entraînement, si l'erreur de validation ne s'améliore plus au bout de *k epochs* (*k* s'appelle la patience), alors on cesse l'entraînement et on restaure les poids optimums trouvés.

Maintenant que nous avons une idée plus précise des caractéristiques des auto-encodeurs utilisés dans la détection d'anomalies, la section suivante s'intéresse à un type de réseaux de neurones capables d'analyser des séries temporelles, plutôt que des valeurs isolées d'un même attribut : il s'agit des réseaux de neurones récurrents (*Recurrent Neural Networks*, RNN).

[29] Notons que cette disposition permet aussi logiquement à l'auto-encodeur exemple d'être utilisé pour de la réduction de dimension ou de l'analyse en composantes principales.

6.2.2 Réseaux de neurones récurrents

Cette section présente d'abord le neurone récurrent puis la couche récurrente pour aboutir au réseau de neurones récurrents. Les applications des RNN série-vers-série sont aussi exposées en regard de mes besoins. Enfin les limitations des RNN sont avancées pour montrer le remplacement progressif des neurones récurrents par des cellules LSTM.

6.2.2.1 Neurone récurrent, couche récurrente

Un neurone récurrent (cf. Figure 6-3), à l'opposé du neurone artificiel présenté précédemment, présente une boucle de rétroaction, ce qui fait que sa sortie actuelle $y[t]$ dépend de ses entrées actuelles $X[t]$ mais aussi de sa sortie passée $y[t\text{-}1]$. Il possède une série de poids W_X liés aux entrées et un poids w_y lié à la sortie passée. L'expression de la sortie actuelle $y[t]$ est :

$$y[t] = \varphi\left(X[t]^T W_X + y[t-1]w_y\right) \tag{6-4}$$

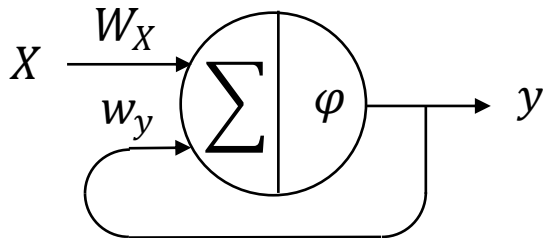

Figure 6-3. Neurone récurrent

En procédant par récurrence, on voit que la sortie du neurone est une fonction des entrées de toutes les étapes précédentes ; par elle, le neurone maintient une forme de mémoire. De manière générale, une partie d'un réseau de neurones qui retient un état au cours des différence étapes temporelles s'appelle cellule de mémoire. Celle procurée par le neurone récurrent est la plus basique. On peut en envisager de plus complètes, par exemple en constituant une couche de neurones implémentant une rétroaction du vecteur Y de sortie vers la partie des entrées dédiée à accueillir les sorties passées. Dans ce cas, chaque neurone aura un vecteur de poids liés aux entrées et un vecteur de poids liés aux sorties passées. La Figure 6-4 montre une couche récurrente de 3 neurones.

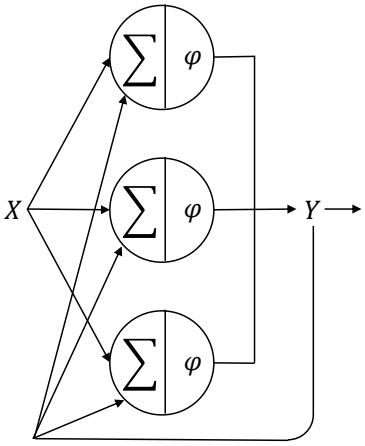

Figure 6-4. Couche récurrente à 3 neurones récurrents

Dans le cas d'un simple neurone récurrent, l'état de la cellule à l'étape *t* est caractérisé par sa sortie *y*[*t*]. Cela peut être plus subtil en dehors de ce cas trivial. En général, on note cet état *H* [*t*] (« H » pour *hidden*, c'est à dire caché), c'est une fonction de l'état précédent *H* [*t*-1] et de l'entrée *X* à l'étape *t* :

$$H[t] = f(H[t-1], X[t]) \tag{6-5}$$

6.2.2.2 Réseaux série vers série

Dans la Figure 6-5 à gauche, le rectangle grisé représente une cellule mémoire. Celle-ci pourrait correspondre au cas d'école du simple neurone récurrent ou, comme je l'ai représenté ici, à une couche de tels neurones (3, ici). En dépliant cette cellule dans le temps selon la taille des séquences sur laquelle on veut travailler (4, ici), on obtient la Figure 6-5 à droite. Cette dernière représentation permet de voir qu'un RNN peut accepter une série en entrée et produire une série en sortie.

Les usages de ce type de réseau « série vers série » (*sequence-to-sequence*) sont principalement ceux liés à la prévision. Par exemple, le réseau peut être entrainé sur des séquences de 4 températures quotidiennes consécutives en entrée et il lui est donné comme séquences de températures de sortie les mêmes mais décalées d'un jour vers le lendemain. Ainsi, le système donne une prévision du futur en apprenant du passé.

Sur cette même Figure 6-5, on imagine sans mal qu'on peut entraîner le réseau pour qu'il apprenne à reconstruire les séquences de RSSI en lui fournissant les mêmes séquences de cette grandeur en entrée et en sortie. On utilise pour cela un auto-encodeur qui ne travaille plus sur des valeurs d'attributs mais sur des séquences d'attributs.

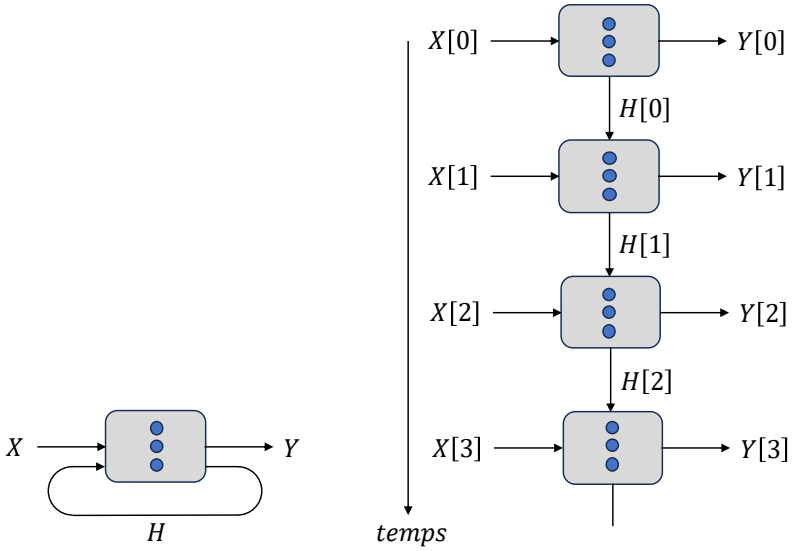

Figure 6-5. Cellule mémoire de 3 neurones (à gauche) et son dépliage sur 4 étapes temporelles (à droite)

Pour passer à un RNN profond, on peut empiler les couches comme on l'avait expliqué avec les ANN en 6.2.1.1, ainsi qu'en témoigne la Figure 6-6.

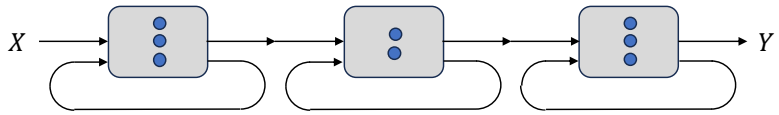

Figure 6-6. Réseau profond de neurones récurrents

Les RNN semblent prometteurs mais ils souffrent de limitations exposées dans la section suivante.

6.2.2.3 Limites des RNN et réseaux de cellules LSTM comme solution

Pour faire travailler les RNN sur de longues séquences, il faut augmenter la taille du dépliage présenté Figure 6-5 mais alors celui-ci devient analogue à une sorte de réseau très profond. Dans ce cas, le problème sous-jacent est celui de l'explosion / disparition de gradient dont le corollaire est un entraînement long et instable. Certes, des solutions pour contrer cette difficulté majeure existent, certaines ont d'ailleurs déjà été évoquées plus tôt, mais elles réclament une bonne expertise.

D'autre part, les RNN ne sont capables que de retenir les entrées d'un passé proche, ne disposant ainsi que d'une mémoire à court terme. Au fur et à mesure qu'ils intègrent de nouvelles entrées, ils oublient le début des séquences, ce qui constitue un obstacle à une bonne modélisation des phénomènes temporels.

Pour pallier à cela, les auteurs des travaux décrits dans (Hochreiter and Schmidhuber, 1997) ont proposé en 1997 la cellule de « mémoire à long et court termes » (*Long Short-Term Memory*, LSTM). On peut se contenter de remplacer les couches de neurones récurrents du RNN par des couches de LSTM, l'ensemble présentera des performances bien meilleures, convergera plus vite et sera capable de détecter les dépendances à long terme. Les LSTM sont utilisés dans les diagnostics médicaux, l'analyse des journaux systèmes et le traitement du langage naturel.

À l'intérieur, une LSTM est assez complexe. Elle maintient deux états cachés, un pour le court terme (*H*) et un pour le long terme (*C*). Une entrée importante est reconnue grâce à une porte d'entrée ; elle est stockée plus ou moins longtemps grâce à une porte d'oubli et elle est extraite quand il le faut grâce à une porte de sortie.

L'unité récurrente à porte (*Gated Recurrent Unit*, GRU) est une variante simplifiée de LSTM tout aussi efficace, ce qui la rend de plus en plus utilisée. Enfin, les réseaux de convolution 1D sont une piste à explorer dans la thématique d'analyse des séquences temporelles.

6.3 Conception d'un IDS détectant les attaques d'usurpation en environnement habité

6.3.1 Introduction : principe de la détection

6.3.1.1 Présentation

Dans cette partie, je me replace dans les conditions de test du détecteur naïf de la section 5.7. Le but ici est d'étudier si, en s'appuyant sur des algorithmes d'apprentissage profond, on peut obtenir une détection plus efficace. Toutes les expériences présentées ici ont été menées hors-ligne.

Nous savons maintenant que les RSSI successifs d'un objet D_j, captés par une sonde P_i, ne sont pas des valeurs indépendantes mais font partie de séquences temporelles. Fort des apports théoriques des sections précédentes, je propose de suivre les séquences de RSSI

d'un objet D_j, captés par une sonde P_i, par un auto-encodeur utilisant des couches de cellules LSTM au lieu de couches de neurones. Avec par exemple 2 sondes et 10 objets, c'est 20 IDS qu'il faut générer !

6.3.1.2 Entraînement

En l'absence d'attaque, pour chaque couple (P_i, D_j), un auto-encodeur est entraîné avec en entrée des séquences de RSSI standardisé. Pendant cet apprentissage, il ajuste au mieux ses poids pour reconstruire en sortie des séquences les plus proches possibles de celles en entrée. À la fin de l'entraînement, un modèle f_{ij} de reconstruction des séquences « normales » est obtenu.

f_{ij} peut être vu comme un **modèle de l'environnement** entre la sonde P_i et l'émetteur légitime D_j. C'est plus subtil qu'une empreinte d'objet.

Pour toute séquence X_{train} d'entrée lié à (P_i, D_j), la séquence de sortie reconstruite $\widehat{X_{train}}$ se calcule par :

$$\widehat{X_{train}} = f_{ij}(X_{train}) \tag{6-6}$$

On désigne par ts le nombre de valeurs dans chaque séquence (*temporal steps*).

On désigne l'erreur de reconstruction de la séquence considérée par $R(X_{train}, \widehat{X_{train}})$.

Nous prenons pour évaluer cette erreur une erreur absolue moyenne sur la séquence[30] :

$$R(X_{train}, \widehat{X_{train}}) = \frac{1}{ts} \sum_{k=0}^{k=ts-1} \left| X_{train}(k) - \widehat{X_{train}}(k) \right| \tag{6-7}$$

Cette formule permettra de tracer la distribution de R à l'entraînement. Celle-ci est utile pour la détermination du seuil d'erreur utilisé dans la détection.

6.3.1.3 Détection

L'auto-encodeur fonctionnant en détection d'anomalies, l'assomption est qu'en test une erreur de reconstruction trop importante entre la séquence de sortie prédite (obtenue par l'application de f_{ij} à la séquence d'entrée) et la séquence d'entrée témoigne d'une anomalie de séquence RSSI de l'objet D_j. C'est le signe que son identifiant est en train d'être usurpé par un nœud malveillant qui ne reproduit pas une séquence normale.

De façon formelle, pour toute séquence X_{test} d'entrée lié à (P_i, D_j), la séquence de sortie reconstruite $\widehat{X_{test}}$ se calcule par :

$$\widehat{X_{test}} = f_{ij}(X_{test}) \tag{6-8}$$

[30] J'ai retenu MAE comme fonction de coût ici plutôt que MSE car les séquences de RSSI ont des dynamiques très importantes, ce qui a une très forte influence sur la MSE.

L'erreur de reconstruction de la séquence est désignée par $R\left(X_{test}, \widehat{X_{test}}\right)$. Elle est évaluée par l'erreur absolue moyenne sur la séquence :

$$R\left(X_{test}, \widehat{X_{test}}\right) = \frac{1}{ts} \sum_{k=0}^{k=ts-1} \left| X_{test}(k) - \widehat{X_{test}}(k) \right| \quad (6\text{-}9)$$

Si elle est supérieure à un certain seuil *error_thresold*, on considère que la séquence est anormale :

$$Si\ R\left(X_{test}, \widehat{X_{test}}\right) \geq error\_threshold\ \Rightarrow Anomalie \quad (6\text{-}10)$$

Le choix de ce seuil est un point délicat du réglage de tout IDS. Si on l'augmente trop, on aura un IDS très tolérant produisant beaucoup de faux négatifs. Au contraire, si on le diminue trop, l'IDS sera très sensible et produira beaucoup de faux positifs. En général, à ce stade de la conception des IDS, l'empirisme est de mise et on ajuste le seuil pour optimiser les métriques de détection. Cette façon de faire n'est pas compatible avec le caractère autonome de l'IDS (et le nombre d'IDS) que je cherche à réaliser. L'obtention non empirique de ce seuil est détaillée un peu plus en avant, en section 6.3.3.

La partie suivante décrit la topologie du réseau utilisé.

6.3.2 Topologie du réseau LSTM retenu

Les travaux des références (Dong et al., 2020), (Lahmadi et al., 2020) et (Madani and Vlajic, 2021) utilisent des réseaux LSTM pour de la détection d'intrusion dans l'IoT. Ces ressources, conjuguées aux ouvrages (Géron, 2019) et (Chollet, 2021) me permettent d'arrêter les choix du Tableau 6-1 pour la topologie du réseau retenu. Je constate que dans des problématiques similaires aux miennes, il n'est pas besoin d'un grand nombre de couches et que le nombre de cellules LSTM par couche est de quelques dizaines.

Paramètre du modèle	**Valeur**
Nombre de couches internes	2
Nombre de LSTM par couche interne	128
Taux de *dropout* sur chaque couche interne	20%
Taille des séquences (notée *ts*)	30

Tableau 6-1. Topologie du réseau LSTM retenu

Concernant la taille des séquences, je choisis, en rapport avec la moyenne glissante sur 30 éléments de l'IDS naïf de la section 5.7, de former des séquences d'également 30 éléments.

Les hyperparamètres de l'algorithme d'apprentissage sont donnés dans le Tableau 6-2.

Hyperparamètres	Valeurs
Optimiseur	Adam
Taux d'apprentissage (valeur par défaut)	0.001
Taille des lots (valeur par défaut)	32
Arrêt précoce Nombre maximum d'*epochs* Patience (en *epochs*) Restauration des poids optimum	Oui 60 4 Oui
Fonction de coût	MAE
Jeu de validation sur jeu bénin	10%
Cadriciel d'apprentissage profond	TensorFlow 2.2.0 / Keras 2.4.3

Tableau 6-2. Hyperparamètres de l'algorithme d'apprentissage

Ne cherchant pas une représentation des données de dimension réduite, mon réseau ne comporte pas d'espace latent. Par contre, on voit sur les Tableau 6-1 et Tableau 6-2 qu'il comporte son ensemble de régularisations pour éviter le surajustement.

6.3.3 Entraînement pour l'obtention d'un modèle de séquences normales de RSSI

L'entraînement devant se faire sur une période sans attaque, j'ai choisi de le réaliser sur toute la journée du 2 juillet 2022 (cf. Figure 5-2 page 83), qui en est dépourvue (cf. 5.4).

On s'intéresse aux trames captées par la sonde RPI2 qui comprennent comme adresse source 16 bits l'identifiant du capteur de mouvement MS1, c'est-à-dire 0x0A12.

Nous effectuons pour ce nouveau cas les trois premières étapes déjà documentées en 5.7 :

1. **Création d'un fichier de captures fusionnées (24h du 2 juillet 2022)**
2. **Filtrage des trames –** Cf. 5.7.2 pour le principe.
3. **Récupération des trames associées à l'identifiant considéré –** 19 287 valeurs de RSSI sont obtenues pour l'identifiant 0x0A12.

Ensuite, les valeurs doivent être standardisées car les cellules LSTM utilisent les fonctions d'activation *Sigmoid* (i.e. *logistic*) et *tanh*, très sensibles à la magnitude de leurs entrées.

4. **Standardisation des valeurs de RSSI –** Les paramètres de la standardisation (moyenne et écart-type) ne doivent être calculés que sur les données du jeu d'entraînement. Si le jeu de test nécessite une standardisation ou sa réciproque, ce

sont celles du jeu d'entrainement[31]. Tous les calculs qui suivent (entraînement, erreurs, etc.) sont faits à partir des valeurs standardisées.

5. **Transformation du jeu d'entraînement de RSSI en un jeu de séquences de RSSI consécutifs –** Avec 19 287 valeurs, on fabrique 19 287 – 30 + 1 soit 19 258 séquences de 30 RSSI consécutifs, se décalant d'un élément à chaque fois par rapport au précédent. Pour l'entraînement, les séquences peuvent être mélangées aléatoirement mais pas les valeurs de RSSI au sein des séquences.
6. **Implémentation du modèle –** Le modèle décrit dans Tableau 6-1 et Tableau 6-2 est implémenté grâce à des classes comme `LSTM` issues de la bibliothèque Python pour l'apprentissage profond nommée Keras (Chollet, 2021).
7. **Entraînement –** Après implémentation, l'entraînement démarre. Il se fait sur un tenseur de rang 3 en entrée et sur le même tenseur en sortie puisqu'on utilise un auto-encodeur. Les dimensions de ces tenseurs selon les axes « nombre de séquences », « taille des séquences » et « nombre d'attributs » sont ici (19 258, 30, 1). Pendant l'entraînement, la surveillance au fil des *epochs* de l'erreur calculée sur un jeu de validation permet de s'assurer que l'algorithme converge.
8. **Choix non empirique du seuil d'erreur pour la future détection –** L'erreur de reconstruction $R\left(X_{train}, \widehat{X_{train}}\right)$ de chaque séquence du jeu d'entraînement est calculée et ses distributions simple et cumulative sont tracées (cf. Figure 6-7 et Figure 6-8). Plusieurs options sont possibles pour le choix de $error\_threshold$. Par exemple, on peut retenir la valeur de R en dessous de laquelle 96% des erreurs se situent. Cela donne une valeur de $error\_threshold$ valant 0.375 (cf. Figure 6-8). Par souci d'autonomie et de simplicité, ce percentile est à conserver pour tous les couples (P_i, D_j) et de ce fait il y a lieu de le choisir de façon optimum, ce qui peut être compliqué. Une autre option est de prendre pour $error\_threshold$ la plus grande des erreurs de reconstruction parmi les séquences d'entraînement. Cela donne une valeur de $error\_threshold$ valant 0.756 (cf. Figure 6-7). Cette option de retenir l'erreur maximum des séquences d'entraînement pour le seuil d'erreur m'a donné des métriques intéressantes même avec des objets très différents de MS1, par exemple la passerelle Zigbee IB1, aussi c'est celle que je retiendrai.

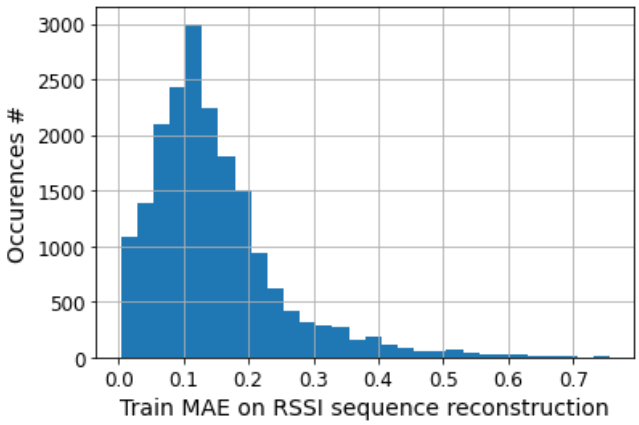

Figure 6-7. Distribution de l'erreur de reconstruction à l'entraînement

[31] Dans le package `sklearn.preprocessing`, on utilise les méthodes de la classe `StandardScaler` pour effectuer les standardisations et leurs réciproques. Pour le jeu considéré, la moyenne vaut -83.95 dBm et l'écart-type 3.26 dBm.

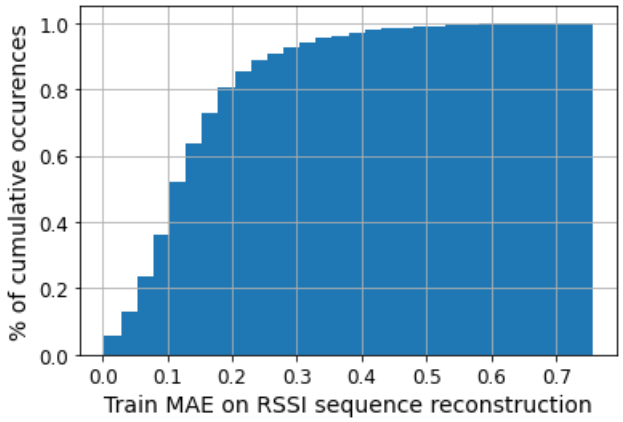

Figure 6-8. Distribution cumulée et normalisée de l'erreur de reconstruction à l'entraînement

Le Tableau 6-3 donne une synthèse des différents paramètres obtenus lors de l'entraînement. Ces paramètres sont également renseignés pour la phase de détection qui fait l'objet de la partie suivante.

		Entraînement	**Détection**
Date		2 juillet 2022 CEST (24 h.)	8 juillet 2022 CEST (24 h.)
Sonde utilisée		RPI2	RPI2
Nombre de trames	Total (cf. 5.7.2)	1 342 431	1 378 154
	Après filtrage (cf. 5.7.2)	787 458	787 481
	Avec identifiant 0x0A12	19 287	20 266
Nombre de séquences de 30 RSSI		19 258	20 237
Seuil d'erreur retenu *error_threshold*		0.756	

Tableau 6-3. Paramètres de l'entraînement et de la détection

6.3.4 Détection des usurpations d'identité

Je me replace dans les conditions du test décrit à la section 5.7 portant sur la journée du 8 juillet 2022. Les RSSI retenus sont ceux des trames capturées par la seule sonde RPI2 et ils doivent avoir l'identifiant 0x0A12 du capteur de mouvement MS1 en adresse source. Il s'agit de voir si un attaquant usurpe l'identité de MS1, à travers les 6 attaques référencées

en 5.7.1 page 91. Les attaques dont la référence finit par « C » sont à 12 trames illégitimes par minute, celles dont la référence finit par « E » sont à 500 trames illégitimes par minute.

La liste des étapes est donnée ci-dessous, les trois premières ont déjà été réalisées en 5.7 :

1. **Création d'un fichier de captures fusionnées (24h du 8 juillet 2022)**
2. **Filtrage des trames –** Cf. 5.7.2.
3. **Récupération des trames associées à l'identifiant considéré –** 20 266 valeurs de RSSI sont obtenues pour l'identifiant 0x0A12.
4. **Standardisation des valeurs de RSSI –** Les paramètres de la standardisation utilisés sont ceux obtenus lors de l'entraînement.
5. **Transformation du jeu de test de RSSI en un jeu de séquences de RSSI consécutifs –** Avec 20 266 valeurs, on fabrique 20 266 – 30 + 1 soit 20 237 séquences de 30 RSSI consécutifs.
6. **Obtention de chaque séquence de sortie reconstruite –** Cela est atteint par application du modèle obtenu à l'entraînement sur chaque séquence d'entrée, cf. Équation (6-8) page 103.
7. **Calcul de $R\left(X_{test}, \widehat{X_{test}}\right)$ l'erreur de reconstruction de chaque séquence –** Cela est obtenu par l'Équation (6-9).
8. **Détermination des séquences qui sont des anomalies –** Cf. Équation (6-10). Quand l'erreur de reconstruction $R\left(X_{test}, \widehat{X_{test}}\right)$ dépasse $error\_threshold$ déterminé à la fin de l'entraînement, alors la séquence est considérée anormale, on considère que ce ne peut être MS1 qui l'a générée et qu'il y a eu usurpation de son identité. 242 anomalies sont recensées. On peut choisir un modèle de génération d'alarmes à partir de celles-ci.

La Figure 6-9 rend compte de cette expérience de détection :

- En haut de la figure, on trouve en orange l'erreur de reconstruction pour toutes les séquences et en bleu l'horizontale ayant pour ordonnée $error\_threshold$, la valeur du seuil d'erreur retenu (0.756). Une anomalie est considérée quand la courbe orange dépasse la courbe bleue.
- En bas de la figure, on trouve en bleu le RSSI instantané associé à l'identifiant 0x0A12. Les anomalies sont portées par des points rouges sur celui-ci. Un zoom temporel permettrait d'en voir 242. La base de vérité des attaques est donnée également : une petite flèche rouge verticale indique à chaque fois le début d'une attaque et sa référence (par exemple « 08C ») est donnée à la suite.

6.3.5 Métriques et résultats

À nouveau, pour élaborer les métriques de détection, je découpe les 24 heures en tranches de 10 minutes pour corréler au sein de chacune d'elles les attaques effectives et les anomalies détectées, afin d'établir les nombres de respectivement, vrais négatifs, vrais positifs, faux négatifs et faux positifs notés respectivement *TN*, *TP*, *FN* et *FP*, puis métriques classiques définies en 2.5.3 : justesse, précision et rappel. Notons qu'il n'y a jamais plus d'une attaque effective par tranche et qu'une seule anomalie suffit à évaluer la tranche comme attaquée. Sur la Figure 6-9 bas, exceptées les occurrences de vrais négatifs, très nombreuses, celles de vrais positifs, faux négatifs et faux positifs sont inscrits en gras par **TP**, **FN** et **FP**. Le Tableau 6-4 page 110 synthétise les résultats tout en rappelant pour comparaison les métriques du détecteur naïf de la section 5.7.

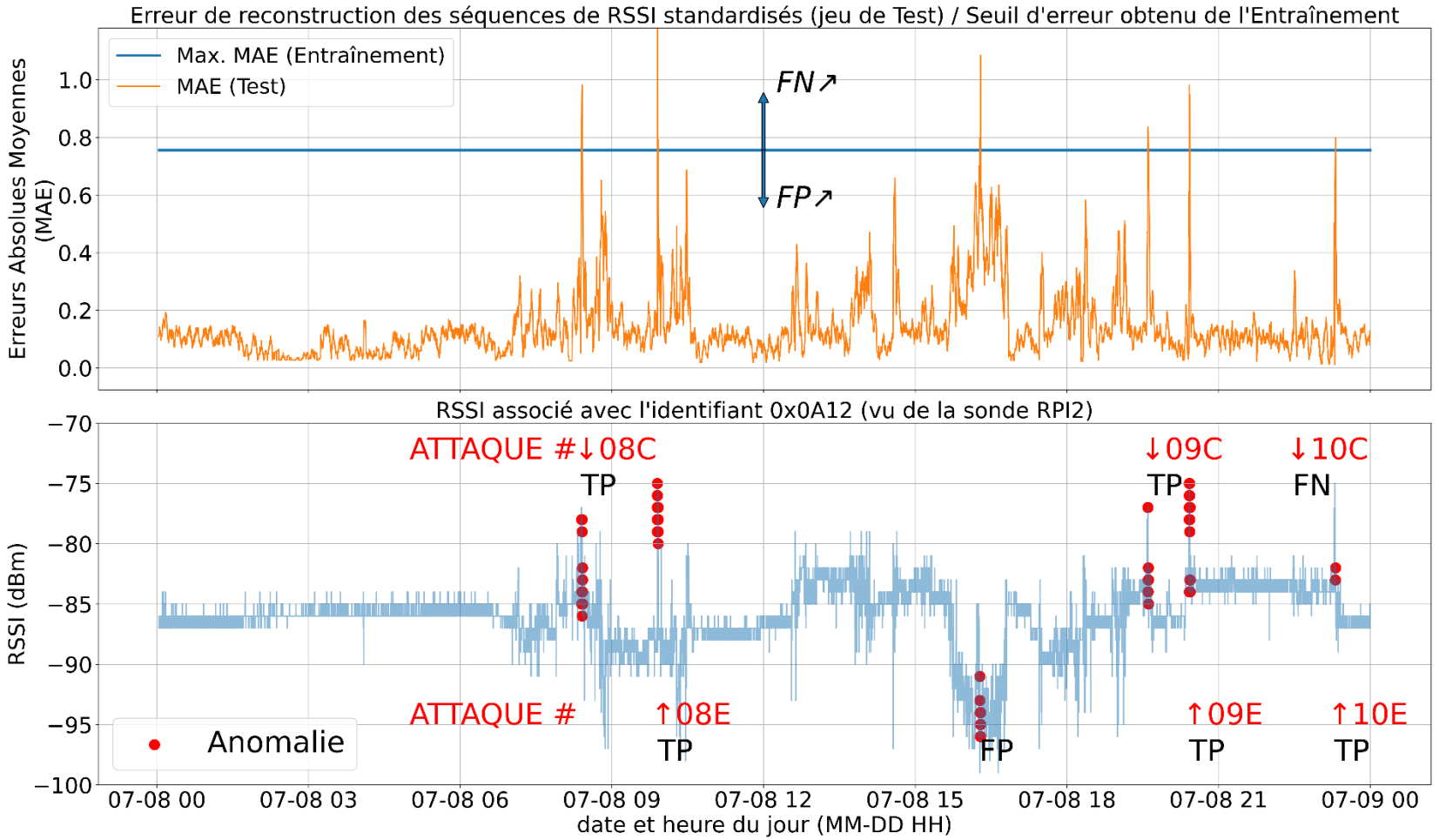

Figure 6-9. Détection d'usurpation d'identité par auto-encodeur à cellules LSTM exploitant des séquences de RSSI (sonde RPI2, identifiant 0x0A12), journée avec 6 attaques du 8 juillet 2022

	TN	*TP*	*FN*	*FP*	*Justesse*	*Précision*	*Rappel*	*TNR*	*FPR*
IDS naïf	127	5	1	11	91.7%	31.2%	83.3%	92.0%	8.0%
IDS LSTM	137	5	1	1	98.6%	83.3%	83.3%	99.3%	0.7%

Tableau 6-4. Comparaison des métriques de détection des deux IDS étudiés

Là encore, le nombre de vrais positifs montre que les attaques sont bien détectées, quelle que soit leur fréquence, petite ou grande. Mais surtout, un net progrès est effectué au niveau des métriques grâce au nombre de faux positifs qui a drastiquement chuté par rapport à la première expérience du détecteur naïf. L'investissement d'ingénierie pour établir ce nouvel IDS est rentable en ce qui concerne la performance de la détection.

Au niveau de l'unique faux positif restant (un peu avant 15:30 sur la Figure 6-9), il semble que le profil du RSSI présenté ici n'a pas été appris par l'IDS pendant la phase d'entraînement. Un apprentissage incrémental permettrait sûrement de transformer ce faux positif en vrai négatif. Je n'ai malheureusement pas eu le temps d'explorer cette piste.

L'unique faux négatif est relatif à l'attaque 10C (8 juillet 2022 à 22:28:10, cf. 5.4 pour un lien vers la liste des attaques) qui ici encore n'est pas repérée. Une investigation me montre qu'un peu avant l'attaque, la valeur moyenne locale du RSSI de l'attaquant s'est rapprochée de la valeur moyenne locale du RSSI de l'objet légitime MS1 (passage de -78 dBm à -85 dBm environ). Cela est probablement dû à un obstacle qui s'est interféré entre l'attaquant et la sonde RPI2 à ce moment. Le résultat est un échappement de la détection. Je pense que l'utilisation de modèles établis en corrélant les données de plusieurs sondes permettra de contourner cette limitation. Une corrélation basique des informations issues des sondes est envisagée à titre de sensibilisation dans la section suivante, dans le cadre d'autres apports.

6.4 Autres apports

6.4.1 Corrélation des informations des sondes : une sensibilisation

Arrivé à ce stade du manuscrit, il est clair que l'utilisation de plusieurs sondes bien placées permet d'améliorer l'efficacité de la détection, notamment en ne ratant plus aucune attaque (plus aucun faux négatif).

On peut envisager un auto-encodeur multivarié à cellules LSTM avec autant d'attributs que de sondes utilisées. Le modèle se construit avec les informations des différentes sondes et pas depuis celles d'une seule. Il s'agit ensuite d'exploiter l'ensemble des erreurs de reconstruction disponibles pour chaque attribut pour signaler le cas échéant des anomalies. Cette proposition me semble prometteuse mais l'architecture est complexifiée : les sondes doivent en permanence communiquer sans fil leurs informations à un IDS central de calcul et ce canal (Wi-Fi ?) introduit une vulnérabilité au sein de l'IDS lui-même[32]. Par ailleurs, il est peut-être nécessaire de resynchroniser les différentes trames captées par les sondes pour un même envoi, afin de les traiter ensemble. Il y a lieu de voir si cette proposition

[32] Dans une habitation respectant les normes NF C 15-100, il y a une prise de communication dans chaque pièce. On peut imaginer que les sondes communiquent par un lien filaire avec l'IDS central, réduisant la vulnérabilité mentionnée.

d'IDS est compatible avec l'IDS réaliste de maison connectée vers lequel je souhaite tendre.

À titre de sensibilisation à la corrélation, je propose ici une structure simple où c'est l'application IDS du smartphone de l'utilisateur qui va effectuer la corrélation des informations de chaque sonde, présentées de façon agrégée. Nous reprenons la journée du 8 juillet 2022, avec deux IDS indépendants comme celui de la section 6.3. Le premier exploite la sonde RPI2 et le second la sonde RPI4.

Le Tableau 6-5 montre le début de chaque tranche horaire de 10 minutes où il y a soit une attaque effective soit une attaque détectée (alors qu'il n'y en a pas eu). Par exemple à 16:13, il n'y avait pas d'attaque mais RPI2 a signalé à tort une anomalie. À 22:33, c'est RPI4 qui a vu une anomalie alors qu'il n'y avait pas d'attaque. Il est dit pour chacune des tranches si RPI2 et RPI4 y voient une (ou plusieurs) anomalie(s).

Début de tranche de 10 min.	**Attaque**	**RPI2 : anomalie(s) ?**	**RPI4 : anomalie(s) ?**	**RPI2 OU RPI4**	**RPI2 ET RPI4**
08:23	8C	Oui	Non	Oui	Non
09:53	8E	Oui	Non	Oui	Non
16:13	–	Oui	Non	Oui	Non
19:33	9C	Oui	Oui	Oui	Oui
20:23	9E	Oui	Oui	Oui	Oui
22:23	10C	Non	Oui	Oui	Non
22:33	–	Non	Oui	Oui	Non
23:13	10E	Oui	Oui	Oui	Oui

Tableau 6-5. Journée du 8 juillet 2022. Attaques et signalement d'anomalies par RPI2 et RPI4. Corrélation des anomalies avec OU et avec ET.

Les anomalies par tranches sont transmises au smartphone par chaque sonde, ce qui représente peu d'information, et celui-ci les corrèle. Le tableau montre une corrélation en OU et une autre en ET. Des cas plus fins peuvent être envisagées (comptage du nombre de Oui par exemple) pour donner une gradation dans l'annonce globale. Par exemple, deux annonces d'anomalies correspondraient plus probablement à une attaque effective qu'une seule annonce.

Les résultats sont plus tranchés qu'avec la sonde RPI2 seule :

- La corrélation en OU donne 6 vrais positifs et 2 faux positifs. Il n'y a plus de faux négatif, ce qui était le but recherché. Le rappel est donc de 100% et la précision de 75%. Cette stratégie de signalement systématique peut intéresser un utilisateur averti (qui souhaite connaître toutes les attaques) et tolérant (aux alarmes).
- La corrélation en ET donne 3 vrais positifs, 2 vrais négatifs et 3 faux négatifs. Le rappel est donc de 50% et la précision de 100%. Cette stratégie rate la moitié des

attaques mais quand elle en signale une à l'utilisateur, celle-ci a véritablement eu lieu, ce qui maintient sa confiance en l'IDS.

Cette corrélation des informations des sondes devrait aussi être envisagée pour détecter les faux positifs, en plus de l'apprentissage incrémental proposé plus tôt.

À titre de curiosité, il aurait été intéressant de pratiquer à la place des corrélations OU et ET un apprentissage supervisé sur les signalements d'anomalies faits par les 4 sondes sur l'ensemble des tranches de 10 minutes de la journée (éventuellement, on peut prendre des tranches plus fines pour avoir plus de points). En label serait donné pour chaque tranche le caractère effectif de l'intrusion (attaque ou pas). La généralisation du modèle obtenu serait à tester sur un autre jour avec attaques. On peut espérer de bons rappels et précisions mais ce n'est pas un système réaliste dans le sens où on l'a entendu jusqu'à maintenant.

6.4.2 Nécessité de choisir des phases d'entraînements représentatives

J'ai pu mettre en évidence que des entraînements non représentatifs des situations de test donnaient pendant la détection un nombre important d'anomalies qui ne correspondaient pas à des attaques effectives. Ainsi par exemple, la combinaison d'un apprentissage réalisé sur quelques heures en situation calme et d'une détection pratiquée sur quelques heures également mais en situation d'activité génère beaucoup de faux positifs. Apprendre sur de longues périodes (plusieurs jours) avec un bon échantillon des situations possibles participerait vraisemblablement à la résolution de ce problème.

6.4.3 Détection d'usurpation appliquée à d'autres types d'objet Zigbee

Je considère les attaques de rejeu de type « C » dans leur usage principal décrit dans le Tableau 5-3. Ici, l'attaquant rejoue des trames que la passerelle internet IB1 (identifiant 0x22FD) avait produites pour contrôler l'ampoule couleur CB2. IB1, *Full Function Device*, présente un profil de prise de parole très différent de MS1, *Reduced Function Device*, utilisé jusqu'à maintenant. Il émet effectivement environ 10 fois plus de trames que MS1 dans la même période. Malgré cette caractéristique, en établissant un modèle de ses profils RSSI en situation bégnine, et en choisissant comme seuil d'erreur le maximum de l'erreur à l'entraînement, les 4 sondes RPI1 à RPI4 parviennent à détecter l'attaque par rejeu 9C du 8 juillet 2022 à 19:35:05 avec 3 heures d'entraînement sur des situations bégnines. Sinon, le test d'usurpation d'identité d'objets-routeurs comme les lampes ou les prises est à envisager.

6.5 Conclusion

Dans ce chapitre, j'ai présenté les réseaux de neurones organisés en auto-encodeur. Ceux-ci permettent, une fois entraînés sur des situations légitimes, de détecter les valeurs anormales dans les attributs considérés. Utilisés avec des neurones récurrents ou des cellules mémoires plus abouties (e.g., LSTM) en lieu et place des neurones, les auto-encodeurs « récurrents », permettent quant à eux de détecter des anomalies dans les séquences d'attributs, pour peu qu'on les ait entraînés sur des séquences bégnines. Ils se prêtent donc bien à l'analyse des séries temporelles de RSSI d'un objet, celui-ci évoluant dans le temps de façon très dynamique quand l'environnement physique entre la sonde et l'objet change régulièrement.

Par la suite, pour exploiter cet attribut volatile dans la détection des attaques d'usurpation, j'ai élaboré et détaillé entièrement un tel auto-encodeur. Lors de l'apprentissage, pour une sonde et un objet donné, un modèle de reconstruction des séquences normales de RSSI est appris. En test, l'hypothèse est faite que les séquences mal reconstruites par le modèle sont

le symptôme d'une anomalie, c'est-à-dire ici d'une usurpation d'identité. La disponibilité du jeu de données utilisé et la description de la méthode suivie permettent de reproduire les expériences proposées voire d'en imaginer d'autres pour valider davantage la solution proposée.

Les résultats de détection de cette solution totalement passive sont nettement meilleurs que ceux du détecteur naïf du chapitre 5, basé sur des seuils de RSSI. Ils montrent que le RSSI, dont l'extraction est aisée, est un candidat de couche physique réaliste pour être exploité dans des algorithmes d'apprentissage profond travaillant à la détection d'intrusion en gérant son caractère volatile. La méthode proposée est à envisager pour pallier simplement une authentification absente (e.g., couche IEEE 802.15.4 de Zigbee) ou pour compléter une authentification jugée peu fiable (e.g., clé de réseau du profil ZLL) en amenant un deuxième facteur d'authentification. Quand une attaque est détectée, une notification peut être envoyée au propriétaire, avec une indication de l'objet dont l'identité est usurpée.

Au niveau des perspectives, la complexité algorithmique de l'établissement des modèles est à évaluer pour voir si cette tâche peut être hébergée dans un hôte restant peu encombrant et de coût modeste. Rappelons que le nombre de modèles à établir et stocker correspond au produit du nombre d'objets à surveiller par le nombre de sondes retenues. On pourra se reporter à la partie 4.3.1.1 page 69 pour plus de détails sur la stratégie de développement envisagée. Pour minimiser le nombre de modèles à établir, j'ai d'abord essayé d'appliquer le modèle établi pour une sonde et un objet aux données RSSI du même objet mais capturées depuis une autre sonde, afin de voir s'il y avait comme une « essence de l'intrusion » dans les données RSSI, indépendamment d'où elles avaient été captées. Les métriques étaient malheureusement peu probantes. Ensuite, mais je n'en ai pas eu le temps, j'aurais aimé voir si le modèle qu'une sonde utilise pour détecter les usurpations d'identité tentées sur un objet donné peut aussi s'appliquer à un autre objet.

La corrélation des informations de plusieurs sondes fait aussi partie des pistes sérieuses à explorer pour combattre l'échappement de la détection et la faible granularité du RSSI. Une corrélation simple a été présentée. Par ailleurs, trouver le nombre optimum de sondes selon la surface à couvrir et l'efficacité de la détection recherchée, sans toutefois augmenter déraisonnablement le coût et la complexité de la solution de sécurité représente également un axe de travail à explorer. Des travaux sur l'estimation de la position des objets grâce à un simple routeur Wi-Fi multi-antennes ont été proposés (Anantharaman et al., 2020) ; il serait intéressant de voir dans quelle mesure ils peuvent être exploités pour étendre notre solution de détection au Wi-Fi. Enfin, je pense que la mise en place d'un apprentissage incrémental contribuerait à diminuer les faux positifs. C'est une hypothèse à valider.

Chapitre 7 - Conclusion et perspectives

En guise de conclusion, je propose d'effectuer d'abord un bilan de la thèse et de ses contributions puis de considérer les limitations et perspectives.

7.1 Bilan des travaux menés

Cette thèse s'est consacrée à l'étude de la sécurité des environnements IoT du type de ceux que l'on rencontre dans les maisons connectées. L'implantation dans les domiciles de réseaux d'objets communicant sans fil est relativement récente et leurs caractéristiques sont très différentes de celles des systèmes traditionnels de traitement de l'information, les exposant à de nombreuses et nouvelles attaques. En effet, la pression économique du marché « maison connectée » étant très forte, celle-ci conduit à des objets à ressources minimales, sans possibilité de leur ajouter de solution de sécurité. Ensuite, il est fait usage de plusieurs piles de protocoles, multipliant les surfaces d'attaques et repoussant une hypothétique standardisation de la sécurité. Enfin, les objets déployés sont administrés par des consommateurs non nécessairement au fait des bonnes pratiques de sécurité. Ceux-ci sont en outre peu enclins à une gestion complexe de leurs objets, qui finalement se retrouvent livrés à eux-mêmes après leur installation initiale. Ces trois raisons concourent à expliquer la faible maturité de la sécurité dans le domaine des domiciles connectés.

Vu ce contexte, je me suis mis en quête des conditions qui rendraient aussi naturelle la diffusion de solutions de sécurité dans l'IoT domestique que l'est celle des antivirus et pare-feux dans l'informatique traditionnelle. Aussi, je visais dans ma thèse à répondre à la problématique suivante, énoncée au Chapitre 3 :

Peut-on protéger les objets contraints des maisons connectées des attaques de communication (comme celles d'usurpation) par des IDS bon marché, universels et ergonomiques ?

La thèse induite nécessitait alors de valider trois hypothèses, chacune traitée dans un chapitre de contribution.

Première hypothèse – *Des considérations globales sur l'écosystème « maison connectée » et les IDS permettent de définir une base d'exigences pertinentes pour la réalisation de solutions de sécurité réalistes et donc diffusables commercialement.*

Dans le Chapitre 4, j'ai proposé une approche de la maison connectée sous un spectre plus large que celui de nombreux travaux, uniquement techniques. Recensant les facteurs économique, technologique et humain façonnant la sécurité de cet environnement, j'ai tenté d'appliquer ces facteurs à la conception d'un IDS afin de dresser une liste d'exigences que celui-ci devait suivre pour avoir une chance d'être adopté dans les foyers. Ainsi, pour ne pas dépasser un coût à l'image de quelques objets, espérer couvrir les technologies typiques d'une maison et favoriser l'adoption par les consommateurs-habitants, j'en suis arrivé à des conclusions dont voici les plus importantes : l'IDS doit être implanté dans un IDS comportemental surveillant passivement le réseau, pas dans les objets, trop contraints et parfois trop fermés. Pour l'écoute passive, il doit utiliser des récepteurs dédiés à une ou plusieurs technologies plutôt qu'une radio logicielle, trop coûteuse et gourmande en ingénierie. Pour la détection d'intrusion, le choix de l'apprentissage non-supervisé avec entraînement sur les échanges légitimes parait le plus réaliste en termes de coût et d'adaptation à des environnements différents. Enfin, des considérations sur l'autonomie de l'IDS ont été menées pour, entre autres points d'ergonomie, s'adapter au côté non-expert de l'utilisateur. Il était important que cette contribution ne reste pas au stade de discussion formelle ne s'appuyant sur aucun élément tangible, auquel cas la validation de l'hypothèse

associée n'aurait pas eu beaucoup de signification. Heureusement, les parties du travail qui ont pu être implémentées ont confirmé plusieurs des choix économiques retenus, notamment l'emploi de récepteurs dédiés pour la capture des échanges ainsi que le choix de l'apprentissage non supervisé, validant par-là, au moins partiellement, la première des hypothèses.

Deuxième hypothèse – *La mise à disposition de jeux de données complet d'échanges intégrant, en plus des données classiques de couche liaison, des attributs de couche physique simples d'accès comme le RSSI permet de favoriser la synthèse d'IDS pouvant détecter entre autres les attaques d'usurpation d'identité.*

Dans le Chapitre 5, j'ai rappelé la nécessité de bénéficier lors de la conception des IDS de jeux de données qualitatifs réalisés dans un contexte représentatif de celui dans lequel les IDS vont travailler : environnement physique, population, variété et types des objets, attaques réalistes. Ces dernières doivent être en outre labellisées afin d'établir les métriques témoignant de l'efficacité des IDS et permettant de comparer ces derniers.

Constatant qu'au sein des technologies IoT, les IDS pour Wi-Fi étaient surreprésentés aussi bien en termes de papiers scientifiques que de jeux de données disponibles, je me suis orienté vers le protocole Zigbee (profil ZLL) pour participer à la fourniture d'outils couvrant des technologies représentatives des maisons connectées. Dans l'état de l'art, j'ai montré que comme les autres protocoles IoT, celui-ci souffre de vulnérabilités, notamment en ce qui concerne l'authentification des objets. Par exemple, celle-ci est absente au niveau de la couche MAC, permettant tout un ensemble d'attaques. Je me suis mis en recherche d'un jeu de données réaliste qui, outre la présence classique des données et métadonnées de couche MAC, contient aussi des attributs de couche physique, permettant de distinguer des instances d'objet. Dans le contexte économique où je m'inscris, le RSSI, attribut de couche physique dépendant de l'emplacement, est un indicateur de choix, car il est accessible facilement dans toutes les technologies via des sondes bon marché, à l'inverse des coûteux attributs radiométriques.

Devant l'absence d'un tel jeu de données, j'ai décidé de créer ZBDS2023, un jeu Zigbee mis à disposition de la communauté pour concevoir des IDS. Capturé pendant 10 jours dans un habitat de 100 m^2 sur deux étages, il fait figurer 10 objets variés placés à leur position dédiée. Toutes les trames échangées sont capturées depuis 4 sondes distribuées, ce qui fournit une redondance au niveau des informations de couche MAC dans les trames capturées par les sondes mais aussi une empreinte de 4 RSSI à chaque fois qu'un objet émet une trame. En outre, 10 sessions de 5 attaques classiques, variées et labellisées parsèment ZBDS2023.

Dès que mon jeu de données a été bâti, un frein à l'expérimentation d'algorithmes et de techniques de visualisation a été levé. Un des premiers apports de ZBDS2023 a été de me faire prendre conscience du caractère très volatile du RSSI lors de sa mesure dans des environnements changeants créés par exemple par des portes qui s'ouvrent et se ferment ou par des habitants qui se déplacent. Ainsi, l'idée d'authentifier un objet par un tuple de RSSI de référence s'est révélée non pertinente, hormis quand la maison se retrouve sans activité, typiquement la nuit. Néanmoins, un cas d'IDS naïf exploitant la moyenne glissante du RSSI mesuré par une seule sonde a été proposé pour détecter les attaques d'usurpation d'identité en la comparant à une paire de seuils. Ce cas d'étude a permis également de décrire la prise en main du jeu de données et de son outillage. Obtenant des métriques honorables, il montre que le RSSI a un rôle à jouer dans la détection économique des attaques d'usurpation d'identité mais qu'il reste vu sa volatilité une grandeur non triviale à appréhender. Tous les résultats mentionnés dans le présent paragraphe montrent que la mise à disposition de jeux de données « multicouche » facilite la création d'IDS pouvant détecter des attaques d'usurpation d'identité et en cela ils valident notre deuxième hypothèse.

Troisième hypothèse – *Des algorithmes d'apprentissage automatique appropriés permettent d'exploiter avec de bonnes métriques de détection un indicateur physique bon marché comme le RSSI, même si celui-ci est volatile en présence des changements d'environnement dus aux activités domestiques.*

Dans le chapitre 6, toujours animé par les problématiques d'authentification et de détection d'attaque d'usurpation d'identité, je ne souhaitais pas renoncer aux côtés économique et universel du RSSI même si j'avais pris conscience de son caractère volatile lors de mes premières utilisations du jeu de données Zigbee. D'un autre côté, il me paraissait inenvisageable d'utiliser l'apprentissage supervisé eu égard au coût humain et financier que représente la labellisation avant l'entraînement et au fait que celle-ci doit être recommencée pour chaque domicile sans possibilité de se passer d'un expert. Dans l'apprentissage non-supervisé, je me suis d'abord orienté vers les techniques de partitionnement (*clustering*) mais je me suis rendu compte que l'exploitation des clusters d'empreintes de RSSI n'était simple que dans les situations statiques et sans attaque. C'est là que j'ai commencé à explorer les algorithmes non-supervisés à entraînement sur les données d'une seule classe, en générale la classe légitime, i.e. celle sans attaque. Le RSSI étant sensible à la modification dynamique des environnements, il parait judicieux de le percevoir comme une série temporelle pour le traiter. Ainsi, l'IDS retenu prit la forme d'un auto-encodeur à cellules LSTM dont toute la mise en œuvre est documentée. Pendant l'entraînement, à garantir sans attaque, il établit pour chaque paire (sonde, objet) un modèle des séquences normales de RSSI. En test, s'il arrive à reconstruire la séquence d'entrée à l'aide du modèle, c'est que la séquence est bégnine. Dans le cas contraire, la séquence témoigne d'une activité malicieuse et l'IDS signale une anomalie correspondant ici à une attaque par usurpation d'identité. Pratiqué sur 24 heures d'entraînement et 24 heures de test, le système obtient des métriques très satisfaisantes qui montre que l'on peut exploiter l'attribut RSSI avec un algorithme d'apprentissage profond pour envisager de pallier (en attendant mieux) une authentification absente ou pour détecter des attaques d'usurpation d'identité, validant ainsi la troisième hypothèse.

Ainsi, dans l'optique de la diffusion commerciale de solutions de protection pour les objets des maisons connectées, le bien-fondé de l'observation holistique de cet environnement dans la phase de conception d'IDS a été démontré. Par ailleurs, l'intérêt de la mise à disposition d'outils de type jeu de données de qualité a été prouvé. Enfin, l'efficacité de certains algorithmes d'apprentissage automatique réussissant à tirer parti d'indicateurs universels bon marché a été validée. Même si mon travail comporte des limitations, exposées plus avant, ces trois conclusions montrent la pertinence de l'approche de la thèse initiale, formulée page 65.

7.2 Limitations et travaux futurs

En ce qui concerne la première contribution (Chapitre 4) :

N'ayant pas à l'issue de ma thèse une implémentation complète de l'IDS recherché, certains points ne peuvent être encore validés. Par exemple, concernant la nécessaire gestion de l'hétérogénéité des protocoles IoT classiques présents dans une maison, je n'ai travaillé qu'avec le protocole Zigbee. J'ai pris soin cela dit de vérifier que le RSSI, attribut principal utilisé dans ce manuscrit, est également disponible de façon simple dans BLE (Lahmadi et al., 2020) et dans Wi-Fi (Anantharaman et al., 2020).

Pour la même raison, des points liés au côté non expert de l'utilisateur ne peuvent être évalués pour l'instant : L'IDS est-il assez autonome ? L'utilisateur a-t-il confiance en ce qu'il annonce ? Son coût de revient est-il assez bas pour déclencher l'achat ? Par ailleurs, installer un IDS qui écoute en permanence et qui va enregistrer des données pose des questions légales relatives au respect de la vie privée.

Achever un prototype pour valider ces deux aspects de l'approche holistique initiale fait partie des perspectives immédiates de ce travail de thèse. On peut noter que la partie « sondes » (cf. 5.5.1) de l'infrastructure de réalisation du jeu de données peut être réexploitée dans l'architecture du prototype d'IDS à réaliser.

En ce qui concerne la seconde contribution (Chapitre 5) :

Nous avons vu que l'usage d'une seule sonde était enclin à favoriser un échappement de la détection par l'attaquant. Celui-ci peut être fortuit mais l'attaquant peut aussi le rechercher volontairement en prenant connaissance dans la documentation de l'objet de sa puissance d'émission, puis en estimant la puissance à la réception si la position de la sonde est connue et enfin en ajustant sa puissance d'émission en conséquence. Pour s'entraîner à détecter de telles stratégies, la partie attaques d'une nouvelle version du jeu de données devrait intégrer une section spécifique où l'attaquant bouge et/ou fait varier sa puissance d'émission. De manière générale, le jeu de données doit évoluer vers l'intégration d'attaques plus intelligentes (Bout et al., 2022) que celles utilisées jusqu'alors.

Par ailleurs, dans la préparation des trames servant d'entrée aux différents IDS envisagés, un script de filtrage (cf. 5.7.2 page 91) élimine plusieurs types de trames, notamment celles présentant une mauvaise somme de contrôle (*bad FCS*, cf. Tableau 2-2 page 46). Selon les fichiers de capture élémentaire d'une heure, le taux de ces trames varie entre 0.4% et 2.0%. Corréler ce taux au placement relatif des sondes et des objets peut participer à un gain d'expérience sur le placement des sondes en intérieur.

En ce qui concerne la troisième contribution (Chapitre 6) :

Mon travail mérite davantage de validation pour assoir les résultats obtenus. J'ai surtout étudié la détection d'usurpation d'identité d'un capteur de mouvement mais c'est un cas d'objet particulier de type feuille dans la topologie réseau, qui fonctionne de surcroit sur batterie. De même, établir un modèle sur une journée bégnine et étudier sa performance sur toutes les journées comprenant au moins une attaque est un travail à effectuer. Enfin, le jeu de données comporte des attaques, notamment de type sybil *Association Request* et de type mascarade *Leave*, dont la détection n'a pas été étudiée (cf. Tableau 5-3 page 90).

Relativement au modèle du réseau LSTM (cf. Tableau 6-1), plusieurs combinaisons de paramètres devraient être testées pour voir leur effet sur les métriques. Dans une moindre urgence, les effets des différents hyperparamètres de l'algorithme d'apprentissage pourraient être analysés.

La complexité algorithmique de l'établissement de plusieurs modèles à base d'auto-encodeurs doit être évaluée afin d'étudier la possibilité d'un accélérateur matériel embarqué dont le coût ne viendrait pas annuler tous les efforts d'économie faits jusqu'ici. S'il y a une possibilité, cela permettrait d'avancer dans la construction d'un prototype dont on pourrait tester l'adoption auprès du grand public.

Au rang des perspectives pures, il serait intéressant de savoir si une détection « multicouche » faisant intervenir le RSSI et d'autres attributs plus classiques de couche MAC amène de la valeur ajoutée dans la détection des attaques étudiées et dans d'autres non abordées dans ce manuscrit. Par exemple, le temps inter-trames et la longueur de trame sont des attributs de couche MAC très classiques pour détecter les dénis de service. Est-ce que l'ajout d'un attribut du type « séquences de RSSI » est possible et, si oui, permettrait-il d'identifier avec plus de certitude l'objet pratiquant un déni de service ou une attaque de l'homme du milieu ? Également, l'utilisation de ces attributs de couches différentes permettrait-elle de limiter le nombre de sondes au domicile ?

Cela dit, les deux perspectives qui nous semblent devoir recevoir de l'attention rapidement sont celles relatives à la diminution des faux-négatifs et à la diminution des faux-positifs.

Le premier objectif est achevable par la corrélation des informations de plusieurs sondes. En effet, les attaques non détectées sont dues soit à un échappement de la détection par l'attaquant qui arrive à imiter le profil RSSI de l'objet légitime, soit à la trop faible granularité du RSSI. Plusieurs sondes permettraient de lever ce souci. Des pistes pour démarrer ont été proposées à la partie 6.4.1. Le deuxième objectif peut à mon avis être atteint par la mise en œuvre d'un apprentissage incrémental sur des données bégnines, par exemple une à deux heures par jour après s'être assuré de l'absence d'un attaquant dans le voisinage. Cela permet au système de s'adapter rapidement à de nouvelles habitudes des habitants et c'est de toute façon plus réaliste que d'« absorber » un lot de données couvrant plusieurs jours d'affilée. Toute l'étude de faisabilité reste cependant à faire.

En conclusion, l'utilisation d'indicateurs de couche physique dans les systèmes de détection d'intrusion des maisons connectées amène une valeur ajoutée à la sécurité de ces environnements tout en respectant les critères économiques, technologiques et humains qui favoriseront la diffusion de ces systèmes.

Bibliographie

Acar, A., Fereidooni, H., Abera, T., Sikder, A.K., Miettinen, M., Aksu, H., Conti, M., Sadeghi, A.-R., Uluagac, S., 2020. Peek-a-boo: i see your smart home activities, even encrypted!, in: Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks. Presented at the WiSec '20: 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM, Linz Austria, pp. 207–218. https://doi.org/10.1145/3395351.3399421

Alrawi, O., Lever, C., Antonakakis, M., Monrose, F., 2019. SoK: Security Evaluation of Home-Based IoT Deployments, in: 2019 IEEE Symposium on Security and Privacy (SP). Presented at the 2019 IEEE Symposium on Security and Privacy (SP), pp. 1362–1380. https://doi.org/10.1109/SP.2019.00013

Anantharaman, P., Song, L., Agadakos, I., Ciocarlie, G., Copos, B., Lindqvist, U., Locasto, M.E., 2020. IoTHound: environment-agnostic device identification and monitoring, in: Proceedings of the 10th International Conference on the Internet of Things, IoT '20. Association for Computing Machinery, Malmö, Sweden, pp. 1–9. https://doi.org/10.1145/3410992.3410993

ANSSI, 2023. Glossaire [WWW Document]. ANSSI. URL https://www.ssi.gouv.fr/entreprise/glossaire/ (accessed 7.24.23).

Apthorpe, N., Reisman, D., Feamster, N., 2017. A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic. ArXiv170506805 Cs.

Ashton, K., 2009. That "Internet of Things" Thing - RFID JOURNAL [WWW Document]. URL https://www.rfidjournal.com/that-internet-of-things-thing (accessed 7.18.23).

Baazaoui, M., Ketata, I., Fersi, G., Fakhfakh, A., Derbel, F., 2022. Implementation of RSSI Module in Omnet++ for Investigation of WSN Simulations based on Real Environmental Conditions. Presented at the Special Session on Energy-Aware Wireless Sensor Networks for IoT, SCITEPRESS, pp. 281–287. https://doi.org/10.5220/0011012600003118

Benkhelifa, E., Welsh, T., Hamouda, W., 2018. A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems. IEEE Commun. Surv. Tutor. 20, 3496–3509. https://doi.org/10.1109/COMST.2018.2844742

Bezawada, B., Bachani, M., Peterson, J., Shirazi, H., Ray, Indrakshi, Ray, Indrajit, 2018. Behavioral Fingerprinting of IoT Devices, in: Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security, ASHES '18. ACM, New York, NY, USA, pp. 41–50. https://doi.org/10.1145/3266444.3266452

Boin, C., Guillaume, X., Grimaud, G., Groléat, T., Hauspie, M., 2022. One Year of DDoS Attacks Against a Cloud Provider: an Overview, in: 2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC). Presented at the 2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC), pp. 1–5. https://doi.org/10.1109/CTISC54888.2022.9849755

Bormann, C., Ersue, M., Keränen, A., 2014. Terminology for Constrained-Node Networks (Request for Comments No. RFC 7228). Internet Engineering Task Force. https://doi.org/10.17487/RFC7228

Bout, E., Loscri, V., Gallais, A., 2022. How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A Survey. IEEE Commun. Surv. Tutor. 24, 248–279. https://doi.org/10.1109/COMST.2021.3127267

Brik, V., Banerjee, S., Gruteser, M., Oh, S., 2008. Wireless device identification with radiometric signatures, in: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, MobiCom '08. Association for Computing Machinery, New York, NY, USA, pp. 116–127. https://doi.org/10.1145/1409944.1409959

Cayre, R., Nicomette, V., Auriol, G., Alata, E., Kaaniche, M., Marconato, G., 2019. Mirage: Towards a Metasploit-Like Framework for IoT, in: 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE). Presented at the 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), pp. 261–270. https://doi.org/10.1109/ISSRE.2019.00034

Chollet, F., 2021. Deep Learning with Python, 2nd ed. Manning.

Cordero, C.G., Vasilomanolakis, E., Wainakh, A., Mühlhäuser, M., Nadjm-Tehrani, S., 2021. On Generating Network Traffic Datasets with Synthetic Attacks for Intrusion Detection. ACM Trans. Priv. Secur. 24, 8:1-8:39. https://doi.org/10.1145/3424155

Cui, A., Stolfo, S.J., 2010. A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan, in: Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC '10. Association for Computing Machinery, New York, NY, USA, pp. 97–106. https://doi.org/10.1145/1920261.1920276

Dadkhah, S., Mahdikhani, H., Danso, P.K., Zohourian, A., Truong, K.A., Ghorbani, A.A., 2022. Towards the Development of a Realistic Multidimensional IoT Profiling Dataset, in: 2022 19th Annual International Conference on Privacy, Security & Trust (PST). Presented at the 2022 19th Annual International Conference on Privacy, Security & Trust (PST), pp. 1–11. https://doi.org/10.1109/PST55820.2022.9851966

Debar, H., Dacier, M., Wespi, A., 1999. Towards a taxonomy of intrusion-detection systems. Comput. Netw. 31, 805–822. https://doi.org/10.1016/S1389-1286(98)00017-6

Dejon, N., 2022. Conception d'un noyau sécurisé pour objets contraints (Thèse de doctorat). Université de Lille.

Dong, S., Li, Z., Tang, D., Chen, J., Sun, M., Zhang, K., 2020. Your Smart Home Can't Keep a Secret: Towards Automated Fingerprinting of IoT Traffic, in: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, ASIA CCS '20. Association for Computing Machinery, New York, NY, USA, pp. 47–59. https://doi.org/10.1145/3320269.3384732

Doshi, R., Apthorpe, N., Feamster, N., 2018. Machine Learning DDoS Detection for Consumer Internet of Things Devices, in: 2018 IEEE Security and Privacy Workshops (SPW). Presented at the 2018 IEEE Security and Privacy Workshops (SPW), pp. 29–35. https://doi.org/10.1109/SPW.2018.00013

Duque, A., Finet, M., Vial, T., Humbert, M., 2021. SDR4IoT BLE & Zigbee RF dataset.

ENISA, 2017. Baseline Security Recommendations for IoT [WWW Document]. URL https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot (accessed 5.9.20).

European Union, 2022. EU Cyber Resilience Act | Shaping Europe's digital future [WWW Document]. URL https://digital-strategy.ec.europa.eu/en/policies/cyber-resilience-act (accessed 7.10.23).

Farahani, S., 2008. ZigBee Wireless Networks and Transceivers, Newnes. ed.

Faria, D.B., Cheriton, D.R., 2006. Detecting identity-based attacks in wireless networks using signalprints, in: Proceedings of the 5th ACM Workshop on Wireless Security, WiSe '06. Association for Computing Machinery, New York, NY, USA, pp. 43–52. https://doi.org/10.1145/1161289.1161298

Galtier, F., Cayre, R., Auriol, G., Kaâniche, M., Nicomette, V., 2020. A PSD-based fingerprinting approach to detect IoT device spoofing. Presented at the 25th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2020).

Géron, A., 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition [Book]. O'REILLY.

Hahm, O., Baccelli, E., Petersen, H., Tsiftes, N., 2016. Operating Systems for Low-End Devices in the Internet of Things: A Survey. IEEE Internet Things J. 3, 720–734. https://doi.org/10.1109/JIOT.2015.2505901

Helluy-Lafont, É., 2021. Sécurité et détection d'intrusion dans les réseaux sans fil (Thèse de doctorat). Université de Lille.

Helluy-Lafont, E., 2019. CVE - CVE-2019-14095 [WWW Document]. URL https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-14095 (accessed 7.12.23).

Helluy-Lafont, E., Boé, A., Grimaud, G., Hauspie, M., 2020. Bluetooth devices fingerprinting using low cost SDR, in: 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC). Presented at the 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), pp. 289–294. https://doi.org/10.1109/FMEC49853.2020.9144756

Hersent, O., Boswarthick, D., Elloumi, O., Soulard, H., 2014. L'internet des objets: les principaux protocoles M2M et leur évolution vers IP avec BACnet, LonWorks, ModBus, KNX, Z-Wave, 6LoWPAN, ZigBee SE 2.0, ETSI M2M... Dunod : L'usine nouvelle, Paris.

Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Homeland Security, 2016. Securing the Internet of Things | Homeland Security [WWW Document]. URL https://www.dhs.gov/securingtheIoT (accessed 7.10.23).

INRIA, 2020. Internet of Things with Microcontrollers: a hands-on course [WWW Document]. FUN MOOC. URL http://www.fun-mooc.fr/en/courses/internet-things-microcontrollers-hands-course/ (accessed 7.28.23).

IoT Analytics, 2023. State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally [WWW Document]. IoT Anal. URL https://iot-analytics.com/number-connected-iot-devices/ (accessed 7.12.23).

Jmila, H., Blanc, G., Shahid, M.R., Lazrag, M., 2022. A Survey of Smart Home IoT Device Classification Using Machine Learning-Based Network Traffic Analysis. IEEE Access 10, 97117–97141. https://doi.org/10.1109/ACCESS.2022.3205023

Jokar, P., Arianpoo, N., Leung, V.C.M., 2013. Spoofing detection in IEEE 802.15.4 networks based on received signal strength. Ad Hoc Netw. 11, 2648–2660. https://doi.org/10.1016/j.adhoc.2013.04.015

Jomaa, N., 2018. Le co-design d'un noyau de système d'exploitation et de sa preuve formelle d'isolation (Thèse de doctorat). Université de Lille.

Kasinathan, P., Pastrone, C., Spirito, M.A., Vinkovits, M., 2013. Denial-of-Service detection in 6LoWPAN based Internet of Things, in: 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Presented at the 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 600–607. https://doi.org/10.1109/WiMOB.2013.6673419

Kaspersky, 2022. 43% of businesses don't protect their full IoT suite [WWW Document]. www.kaspersky.com. URL https://www.kaspersky.com/about/press-releases/2022_43-of-businesses-dont-protect-their-full-iot-suite (accessed 7.13.23).

Kelly, R.L., 2020. H.R.1668 - 116th Congress (2019-2020): IoT Cybersecurity Improvement Act of 2020 [WWW Document]. URL http://www.congress.gov/bill/116th-congress/house-bill/1668 (accessed 7.10.23).

Kolias, C., Kambourakis, G., Stavrou, A., Voas, J., 2017. DDoS in the IoT: Mirai and Other Botnets. Computer 50, 80–84. https://doi.org/10.1109/MC.2017.201

Kolias, C., Stavrou, A., Voas, J., Bojanova, I., Kuhn, R., 2016. Learning Internet-of-Things Security "Hands-On." IEEE Secur. Priv. 14, 37–46. https://doi.org/10.1109/MSP.2016.4

La, Q.D., Nguyen-Nam, D., Ngo, M.V., Hoang, H.T., Quek, T.Q.S., 2018. Dense Deployment of BLE-Based Body Area Networks: A Coexistence Study. IEEE Trans. Green Commun. Netw. 2, 972–981. https://doi.org/10.1109/TGCN.2018.2859350

Lahmadi, A., Duque, A., Heraief, N., Francq, J., 2020. MitM Attack Detection in BLE Networks using Reconstruction and Classification Machine Learning Techniques. Presented at the MLCS 2020 - 2nd Workshop on Machine Learning for Cybersecurity. https://doi.org/10.1007/978-3-030-65965-3_10

Lee, C., Zappaterra, L., Choi, K., Choi, H.-A., 2014. Securing smart home: Technologies, security challenges, and security requirements, in: 2014 IEEE Conference on Communications and Network Security. Presented at the 2014 IEEE Conference on Communications and Network Security, pp. 67–72. https://doi.org/10.1109/CNS.2014.6997467

Lee, T.-H., Wen, C.-H., Chang, L.-H., Chiang, H.-S., Hsieh, M.-C., 2014. A Lightweight Intrusion Detection Scheme Based on Energy Consumption Analysis in 6LowPAN, in: Huang, Y.-M., Chao, H.-C., Deng, D.-J., Park, J.J. (Jong H. (Eds.), Advanced Technologies, Embedded and Multimedia for Human-Centric Computing, Lecture Notes in Electrical Engineering. Springer Netherlands, Dordrecht, pp. 1205–1213. https://doi.org/10.1007/978-94-007-7262-5_137

Lourme, O., 2020a. Low Power Wide Area Network [WWW Document]. Wikipédia. URL https://fr.wikipedia.org/w/index.php?title=Low_Power_Wide_Area_Network&oldid=204500139 (accessed 7.10.23).

Lourme, O., 2020b. Magazine Programmez!, Numéros 237-238 , Février 2020, “ESP32, Mongoose OS et GCP Cloud IoT Core : un trio efficace et sûr pour l’IoT” [WWW Document]. URL https://bit.ly/2vr7Jkf

Lourme, O., Grimaud, G., Hauspie, M., 2023. ZBDS2023: A multi location Zigbee dataset to build innovative IoT Intrusion Detection Systems, in: 2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Presented at the 2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Montréal, Canada, pp. 84–91. https://doi.org/10.1109/WiMob58348.2023.10187745

Lourme, O., Hauspie, M., 2023a. ZBDS2023 Zigbee dataset. https://doi.org/10.57745/NDW74U

Lourme, O., Hauspie, M., 2023b. Détection des attaques d’usurpation dans les maisons connectées par analyse du RSSI, in: RESSI 2023 : Rendez-Vous de La Recherche et de l’Enseignement de La Sécurité Des Systèmes d’Information.

Lourme, O., Hauspie, M., 2021a. Contribution à l’adoption des IDS dans l’IoT, in: Journée thématique du 11 mai 2021 du Groupe de Travail Sécurité des Systèmes, Logiciels et Réseaux (GT SSLR).

Lourme, O., Hauspie, M., 2021b. Toward a realistic Intrusion Detection System dedicated to smart-home environments, in: 2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Presented at the 2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), virtual, pp. 80–85. https://doi.org/10.1109/WiMob52687.2021.9606337

Madani, P., Vlajic, N., 2021. RSSI-Based MAC-Layer Spoofing Detection: Deep Learning Approach. J. Cybersecurity Priv. 1, 453–469. https://doi.org/10.3390/jcp1030023

Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N.O., Elovici, Y., 2017. ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis, in: Proceedings of the Symposium on Applied Computing, SAC ’17. Association for Computing Machinery, New York, NY, USA, pp. 506–509. https://doi.org/10.1145/3019612.3019878

Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.-R., Tarkoma, S., 2017. IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). Presented at the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 2177–2184. https://doi.org/10.1109/ICDCS.2017.283

Mitchell, R., Chen, I.-R., 2014. A survey of intrusion detection in wireless network applications. Comput. Commun. 42, 1–23. https://doi.org/10.1016/j.comcom.2014.01.012

MITRE, 2022. CVE - CVE [WWW Document]. URL https://cve.mitre.org/ (accessed 7.13.23).

Mordor Intelligence, 2022. ZigBee Market Size & Share Analysis - Industry Research Report - Growth Trends [WWW Document]. URL https://www.mordorintelligence.com/industry-reports/zigbee-market (accessed 7.25.23).

Nguyen, N.T., Zheng, G., Han, Z., Zheng, R., 2011. Device fingerprinting to enhance wireless security using nonparametric Bayesian method, in: 2011 Proceedings

IEEE INFOCOM. Presented at the 2011 Proceedings IEEE INFOCOM, pp. 1404–1412. https://doi.org/10.1109/INFCOM.2011.5934926

Nordic Semiconductors, 2019. Nordic Semiconductor Infocenter [WWW Document]. URL https://infocenter.nordicsemi.com/index.jsp?topic=/com.nordic.infocenter.thread_zigbee.v3.0.0/zigbee_hw_and_mem.html (accessed 7.18.23).

Oh, D., Kim, D., Ro, W.W., 2014. A Malicious Pattern Detection Engine for Embedded Security Systems in the Internet of Things. Sensors 14, 24188–24211. https://doi.org/10.3390/s141224188

OWASP, 2018. OWASP Internet of Things Project - OWASP [WWW Document]. URL https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project (accessed 7.20.20).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Radanliev, P., De Roure, D., Cannady, S., Montalvo, R.M., Nicolescu, R., Huth, M., 2018. Economic impact of IoT cyber risk - Analysing past and present to predict the future developments in IoT risk analysis and IoT cyber insurance, in: Living in the Internet of Things: Cybersecurity of the IoT - 2018. Presented at the Living in the Internet of Things: Cybersecurity of the IoT - 2018, pp. 1–9. https://doi.org/10.1049/cp.2018.0003

Raza, S., Wallgren, L., Voigt, T., 2013. SVELTE: Real-time intrusion detection in the Internet of Things. Ad Hoc Netw. 11, 2661–2674. https://doi.org/10.1016/j.adhoc.2013.04.014

Ren, J., Dubois, D.J., Choffnes, D., Mandalari, A.M., Kolcun, R., Haddadi, H., 2019. Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach, in: Proceedings of the Internet Measurement Conference. Presented at the IMC '19: ACM Internet Measurement Conference, ACM, Amsterdam Netherlands, pp. 267–279. https://doi.org/10.1145/3355369.3355577

Riquet, D., 2015. DISCUS : une architecture de détection d'intrusions réseau distribuée basée sur un langage dédié (These de doctorat). Lille 1.

Ronen, E., Shamir, A., 2016. Extended Functionality Attacks on IoT Devices: The Case of Smart Lights, in: 2016 IEEE European Symposium on Security and Privacy (EuroS P). Presented at the 2016 IEEE European Symposium on Security and Privacy (EuroS P), pp. 3–12. https://doi.org/10.1109/EuroSP.2016.13

Ronen, E., Shamir, A., Weingarten, A.-O., O'Flynn, C., 2017. IoT Goes Nuclear: Creating a ZigBee Chain Reaction, in: 2017 IEEE Symposium on Security and Privacy (SP). Presented at the 2017 IEEE Symposium on Security and Privacy (SP), pp. 195–212. https://doi.org/10.1109/SP.2017.14

Rosay, A., 2022. Détection d'intrusions dans les objets connectés par des techniques d'apprentissage automatique : étude dans les domaines de l'éducation et des voitures connectées (These de doctorat). Le Mans.

Roux, J., Alata, E., Auriol, G., Kaâniche, M., Nicomette, V., Cayre, R., 2018. RadIoT: Radio Communications Intrusion Detection for IoT - A Protocol Independent Approach, in: 2018 IEEE 17th International Symposium on Network Computing

and Applications (NCA). Presented at the 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), pp. 1–8. https://doi.org/10.1109/NCA.2018.8548286

Sadikin, F., Deursen, T. van, Kumar, S., 2020. A ZigBee Intrusion Detection System for IoT using Secure and Efficient Data Collection. Internet Things 12, 100306. https://doi.org/10.1016/j.iot.2020.100306

Shahid, M.R., Blanc, G., Jmila, H., Zhang, Z., Debar, H., 2020. Generative Deep Learning for Internet of Things Network Traffic Generation, in: 2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC). Presented at the 2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 70–79. https://doi.org/10.1109/PRDC50213.2020.00018

Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A.A., 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy. Presented at the 4th International Conference on Information Systems Security and Privacy, SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal, pp. 108–116. https://doi.org/10.5220/0006639801080116

Siby, S., Maiti, R.R., Tippenhauer, N.O., 2017. IoTScanner: Detecting Privacy Threats in IoT Neighborhoods, in: Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, IoTPTS '17. Association for Computing Machinery, New York, NY, USA, pp. 23–30. https://doi.org/10.1145/3055245.3055253

Silabs, 2021. UG103.2: Zigbee Fundamentals [WWW Document]. URL https://www.silabs.com/documents/public/user-guides/ug103-02-fundamentals-zigbee.pdf

Stelte, B., Rodosek, G.D., 2013. Thwarting attacks on ZigBee - Removal of the KillerBee stinger, in: Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013). Presented at the 2013 9th International Conference on Network and Service Management (CNSM), IEEE, Zurich, Switzerland, pp. 219–226. https://doi.org/10.1109/CNSM.2013.6727840

Threatpost, 2021. IoT Attacks Skyrocket, Doubling in 6 Months [WWW Document]. URL https://threatpost.com/iot-attacks-doubling/169224/ (accessed 7.12.23).

Tournier, J., 2021. Modélisation de réseaux IoT hétérogènes à des fins d'évaluation de sécurité (phdthesis). Université de Lyon.

Tschofenig, H., Baccelli, E., 2019. Cyberphysical Security for the Masses: A Survey of the Internet Protocol Suite for Internet of Things Security. IEEE Secur. Priv. 17, 47–57. https://doi.org/10.1109/MSEC.2019.2923973

Usine Digitale, 2018. L'Internet des objets, une opportunité de 151 milliards de dollars en 2018 et 1 567 milliards en 2025 [WWW Document]. URL https://www.usine-digitale.fr/article/l-internet-des-objets-une-opportunite-de-151-milliards-de-dollars-en-2018-et-1-567-milliards-en-2025.N734709 (accessed 7.12.23).

Wallgren, L., Raza, S., Voigt, T., 2013. Routing Attacks and Countermeasures in the RPL-Based Internet of Things. Int. J. Distrib. Sens. Netw. 9, 794326. https://doi.org/10.1155/2013/794326

Xu, Q., Zheng, R., Saad, W., Han, Z., 2016. Device Fingerprinting in Wireless Networks: Challenges and Opportunities. IEEE Commun. Surv. Tutor. 18, 94–104. https://doi.org/10.1109/COMST.2015.2476338

Zarpelão, B.B., Miani, R.S., Kawakani, C.T., de Alvarenga, S.C., 2017. A survey of intrusion detection in Internet of Things. J. Netw. Comput. Appl. 84, 25–37. https://doi.org/10.1016/j.jnca.2017.02.009

Zillner, T., 2016. ZigBee exploited - The good, the bad and the ugly [WWW Document]. Magdebg. J. Zur Sicherheitsforschung. URL http://www.sicherheitsforschung-magdeburg.de/uploads/journal/MJS_045_Zillner_ZigBee.pdf