Objets connectés et vie privée : le long chemin restant

Mathieu Thiery, Emmanuel Baccelli, Freie Univ, Aurélien Berlin

Université Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Mathieu THIERY

Thèse dirigée par **Vincent ROCA, Chargé de Recherches, Centre Inria – Université Grenoble Alpes**, et
codirigée par **Arnaud LEGOUT, Chargé de Recherches, Inria Sophia-Antipolis Méditerranée, Université Côte d'Azur**

préparée au sein de **Institut National de Recherche en Informatique et en Automatique**
dans **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Objets connectés et vie privée : le long chemin restant

Connected devices and privacy: a long journey ahead

Thèse soutenue publiquement le **14 décembre 2020**,
devant le jury composé de :

Monsieur Emmanuel BACCELLI, Professeur Freie Univ. Berlin, rapporteur
Monsieur Aurélien FRANCILLON, Assistant Professeur EURECOM, rapporteur
Monsieur Luc BOUGANIM, Directeur de Recherche, examinateur
Monsieur, Didier DONSEZ, Professeur Univ. Grenoble Alpes, examinateur et **président du jurry**
Monsieur Vincent ROCA, Chargé de Recherches, directeur de thèse
Monsieur Arnaud LEGOUT, Chargé de Recherches, co-directeur de thèse

Table des matières

1 Introduction **8**
- 1.1 Contexte . . . 8
- 1.2 Motivations . . . 10
- 1.3 Contributions et organisation de la thèse . . . 10
- 1.4 Publications . . . 11

2 État de l'art **13**
- 2.1 Objets connectés . . . 13
  - 2.1.1 Contexte et définitions . . . 13
  - 2.1.2 Panorama des objets connectés . . . 14
  - 2.1.3 Acteurs et modèle économique . . . 17
- 2.2 Protocoles et connectivité . . . 18
  - 2.2.1 Modèle OSI . . . 19
  - 2.2.2 Caractéristiques . . . 21
- 2.3 Vie privée . . . 25
  - 2.3.1 Panorama . . . 25
  - 2.3.2 Acquisition de données, découverte d'appareils, fuites et inférences 26
  - 2.3.3 Usage non conventionnel des systèmes . . . 27
  - 2.3.4 Analyse automatique des applications et interaction réseau . . . 27
  - 2.3.5 Défenses . . . 28
  - 2.3.6 Consentement et comportements utilisateurs . . . 29
- 2.4 Sécurité . . . 29
  - 2.4.1 Panorama . . . 29
  - 2.4.2 Vulnérabilités des objets, applications, protocoles et technologies de connectivité . . . 30
  - 2.4.3 Injection de commandes . . . 31
  - 2.4.4 Défenses . . . 31
- 2.5 Autres considérations non traitées . . . 31
- 2.6 Positionnement de cette thèse . . . 32

3 Allumer et éteindre une ampoule : des conséquences pas si anodines **33**
- 3.1 Introduction . . . 33
- 3.2 Méthodologie . . . 34
  - 3.2.1 Buts du travail . . . 34
  - 3.2.2 Environment de test . . . 35
  - 3.2.3 Captures et méthodologie . . . 37
  - 3.2.4 Les 8 scénarios de références et 33 scénarios d'usage . . . 38

3.2.5 Nos outils de statistiques et labellisation . . . 39
3.2.6 Accéder aux données en clair des paquets . . . 39
3.2.7 Préoccupations éthiques et reproductibilité . . . 40
3.3 Résultats expérimentaux . . . 40
3.3.1 Taxonomie des chemins de données . . . 40
3.3.2 Implications en matière de souveraineté . . . 45
3.3.3 Inférence d'action utilisateur via l'analyse de la taille de la requêtes ON/OFF . . . 48
3.3.4 Fuite de données dans le contenu des messages . . . 50
3.3.5 Les traqueurs d'applications . . . 52
3.3.6 Caractéristiques bénéfiques pour la vie privée . . . 52
3.4 Rappel des travaux connexes et contributions . . . 55
3.5 Conclusion . . . 56
3.6 Divulgation responsable . . . 57

4 Obtention de consentement : analyse des configurations et chartes de vie privée 58
4.1 Introduction . . . 58
4.2 Configuration des applications et objets con-nectés . . . 59
4.2.1 Synthèse des problèmes rencontrés . . . 59
4.2.2 Cas des trois applications officielles : un consentement fragile . 60
4.2.3 Des applications tierces qui prennent trop de libertés . . . 64
4.2.4 Concentrateurs de services . . . 67
4.2.5 Responsabilité des fabricants . . . 71
4.3 Chartes de vie privée et consentement . . . 71
4.3.1 Le besoin d'un consentement utilisateur . . . 71
4.3.2 Au sujet des chartes de vie privée . . . 71
4.4 Proposition de solution . . . 74

5 Conclusion et perspectives 81
5.1 Vie privée : une affaire complexe, quelles solutions ? . . . 81
5.2 Paradoxe de la recherche en vie privée . . . 83
5.3 Données simples, à caractère personnel et sensibles . . . 84
5.4 Consentement et outils communs . . . 85
5.5 Utilisateurs multiples et passant . . . 86
5.6 Internet des objets et écologie . . . 86

Résumé

À l'ère de l'Internet des Objets (IdO), les maisons intelligentes sont de plus en plus équipés d'objets connectés, dont les implications sur la vie privée restent largement inconnues des utilisateurs finaux. Dans ce travail, je présenterai les problèmes qu'un utilisateur peut rencontrer lors de l'installation et de l'utilisation d'objets connectés, en focalisant mon étude sur des ampoules intelligentes représentatives et populaires venant de différents fabricants. J'analyserai les implications sur la vie privée des diverses techniques de contrôle de ces ampoules (c.-à-d., l'allumage et l'extinction de l'ampoule à l'aide de différentes applications smartphone, enceintes connectées et boutons intelligents). Je montrerai que les problèmes liés à la vie privée dans le contexte des maisons intelligentes dépendent en grande partie de la technique de contrôle adoptée par l'utilisateur pour contrôler un objet connecté, et non pas seulement l'objet lui-même. Je montrerai qu'il existe des problèmes de souveraineté, de fuites de données, d'inférences de comportements, de traçage, ou encore de conception des applications. Je montrerai dans un second temps les problèmes liés aux configurations des applications et aux chartes de vie privée et montrerai que les fabricants étudiés ne peuvent pas prétendre recueillir un consentement valide de l'utilisateur. Je soulignerai que, dans la plupart des cas, l'utilisateur final ne peut pas raisonnablement comprendre la situation qui est trop complexe (protocoles différents, quantité de données partagées, manque de connaissance). Par conséquent, la situation actuelle n'est ni respectueuse de la vie privée de l'utilisateur, ni légale (pas de consentement), ni souveraine (dépendances avec des acteurs étrangers). Je proposerai donc quelques pistes pour résoudre ces problèmes, notamment en préconisant l'utilisation d'objets connectés et applications fonctionnant localement, et en suggérant l'utilisation de tables de consentement afin de définir simplement ce à quoi l'utilisateur consent exactement.

mots-clés— smartphone, objets connectés, fuites de données, vie privée, consentement

Abstract

In the IoT era, smart homes are getting increasingly equiped with connected devices, whose implications on privacy remain largely unknown from end users. In this work, I will present the issues that a user can face when installing and using connected devices, by focusing my study on smart bulbs comming from different manufacturers, which are popular and representative. I will analyse the privacy implications of multiple control techniques used for these bulbs (i.e., switching on and off the bulb, using different smartphone applications, smart speakers and smart buttons). I will show that privacy issues in the context of smart homes depend mostly on the control technique chosen by the user to control his device, rather than the device itself. I will show that there are issues concerning sovereignty, data leaks, behaviour inferences, tracking, or even applications design. I will then show the issues concerning application configurations and privacy policies and demonstrate that manufacturers cannot pretend to collect a valid consent from users. I will highlight that, in most cases, the end user cannot reasonably understand the situation which is too complex (different protocols, volume of shared data, lack of knowledge). Hence, the current situation is neither privacy-preserving, nor legal (no consent), nor sovereign (dependancy on remote actors). Therefore, I will sugest some ideas to solve these issues, like using connected devices and applications that can work locally, or creating tables of consent in order to easily define what a user consents to exactly.

keywords— smartphone, connected devices, data leakage, privacy, consent

Remerciements

Je tiens à remercier bon nombre de personnes sans qui cette thèse n'aurait certainement pas vu le jour.

Tout d'abord Dr. Vincent ROCA et Dr. Arnaud LEGOUT pour avoir dirigé ma thèse et pour avoir été toujours disponibles pour moi. Ils ont toujours su me soutenir et me fournir des conseils précieux malgré mon caractère parfois opposant.

Je remercie aussi Dr. Aurélien FRANCILLON, Dr. Emmanuel BACCELLI, Dr. Didier DONSEZ et Dr. Luc BOUGANIM d'avoir accepté d'examiner les travaux regroupés dans ce document et de faire partie de mon jury.

Ensuite l'équipe Privatics, dans laquelle règne une ambiance que je n'ai pas vu ailleurs. Entraide et bienveillance sont toujours au rendez-vous et j'ai vraiment apprécié de travailler avec chacun de ses membres. Merci en particulier à M. Levent DEMIR pour les opportunités qu'il m'a offertes, et à M. Belkacem TEIBI pour m'avoir apporté la voix de la sagesse mainte fois malgré mes bizarreries perpétuelles.

Enfin je remercie mes amis et ma famille. Un merci particulier, d'une part à ma mère pour m'avoir toujours aidé à prendre les bonnes décisions et avoir tout donné pour que je bénéficie d'une scolarité qu'elle n'a pas eu la chance d'avoir, et d'autre part à mon père pour m'avoir soutenu dans mon apprentissage et inculqué l'amour de la science et la soif de savoir. Cette thèse est aussi l'aboutissement de leurs efforts conjoints.

Merci à tous.

À mes parents.

Chapitre 1

Introduction

1.1 Contexte

Au commencement de ma thèse, en 2017, j'étais tombé sur un article de Schneier [1], qui selon moi reflète bien la situation dans laquelle nous étions et, dans une certaine mesure, dans laquelle nous sommes encore. Dans son essai, Schneier considérait qu'il était temps de se rendre compte que « nous n'avons plus des objets embarquant des ordinateurs, mais des ordinateurs attachés à des objets ». Il y exprimait ses inquiétudes vis-à-vis de la sécurité et de la vie privée et donnait des exemples de lignes directrices à suivre dans le but d'améliorer la situation.

La vision de Schneier est toujours d'actualité. De plus en plus d'objets connectés arrivent sur le marché. En 2017, au début de cette thèse, on estimait le nombre d'objets connectés à 8.4 milliards [2]. En 2020, ce nombre dépassait les 20 milliards [3]. Non seulement leur nombre augmente significativement, mais leur usage est également plus courant. En 2016 déjà, l'accès à Internet via les smartphones était plus fréquent que via les ordinateurs [4] et, dès 2019, on estimait que 20% des interactions avec ces smartphones passaient par les assistants virtuels personnels (AVP) type Google Assistant, Alexa d'Amazon ou Siri d'Apple [5].

Cette compétition qui consiste à être présent sur de plus en plus d'appareils, dans de plus en plus de lieux et d'occasions, prend maintenant place également dans des sphères particulières qui ont fait leur apparition plus récemment. Les meilleurs exemples se trouvent dans les AVP qui s'installent dorénavant directement dans les foyers grâce, notamment, aux enceintes connectées type Google Home, Amazon Echo ou Apple HomePod. Une guerre est déclarée entre les grands acteurs et, là ou Amazon avait le monopole dès le départ, Google a réussi un tour de force en les dépassant en 2018 [6].

D'autres appareils s'adaptent à cette mode du connecté : les jouets, les thermostats, les montres, tout y passe. Le modèle économique que les fabriquants suivent, les contraint à vouloir s'incruster toujours plus, jusque dans la chambre des enfants [7].

Malheureusement, ces innovations ne sont pas sans conséquences et font régulièrement l'objet de scandales. Des AVP qui envoient [8] ou écoutent [9] vos conversations sans votre consentement, ou qui exécutent des commandes à tort à des fins publicitaires [10] ou commerciales [11], des jouets pour enfants qui les espionnent [12], ou encore des aspirateurs qui cartographient votre foyer afin d'améliorer leur algorithme de parcours dans les différentes pièces de la maison, mais aussi pour mettre aux enchères la liste de vos meubles [13]. Les applications ne sont pas en manque non plus. En 2017, 7 applications sur 10 partageaient leurs données avec des parties tierces [14].

Comme on le voit ici, ces objets connectés sont très loin de garantir que votre vie privée sera préservée. Mais votre sécurité non plus. Les failles de sécurité sont omniprésentes dans

ce milieu et peuvent concerner tout le monde, enfants compris. Amazon s'était d'ailleurs engagé à ne pas vendre de jouet vulnérables [15], mais ce faisant se substitue au régulateur qui édicte les normes, dont celles de sécurité, ce qui révèle un problème dans la façon dont le respect de ces normes est imposé en amont. Beaucoup d'objets connectés n'ont quasiment aucune sécurité, mais les AVP sont une exception de ce point de vue car ils sont, pour la plupart, fabriqués par de grosses sociétés ayant des ressources majeures leur permettant de bien protéger leur système. Cependant, même avec tous ces efforts déployés, ils n'échappent pas à certains menaces [16]. Mais pire encore, ces vulnérabilités pourraient engendrer des risques pour votre vie, par exemple lorsque l'objet attaqué est un véhicule dans lequel vous vous trouvez [17].

Ces failles étant partout, elles peuvent servir de medium lors d'une attaque de masse type DDoS, comme l'a montré le botnet Mirai [18]. Les objets connectés sont également facilement accessibles et analysables. C'est pourquoi des outils comme Shodan [19] ont été créés, permettant de lister ces objets, voire d'en découvrir automatiquement les failles.

Les objets sont aussi dépendants des services qui leur donne vie. Aussi lorsque ces services cessent de fonctionner soit suite à un problème technique, soit parce que l'entreprise qui les fournissait décide d'arrêter, ces objets deviennent inutilisables. Un exemple alarmant s'est produit lors de l'instauration du Règlement Général sur la Protection des Données (RGPD) [20], qui a bouleversé toutes les entreprises et à créé des pannes inattendues [21], du fait de l'incompatibilité de certains objets connectés ou services avec le RGPD. Par exemple, les ampoules Yeelight ont cessé de fonctionné, et Razer a arrêté l'accès à l'un de ses logiciels dans le cloud.

On voit bien que si ce genre d'arrêt se produit sur un pacemaker, sur une caméra de sécurité, ou sur une montre intelligente, cela n'aura pas le même impact. Le fait que certains produits cessent de fonctionner est également souvent lié au modèle économique des entreprises qui les fabriquent. En effet, la collecte massive de données reste une solution très adoptée dans le contexte actuel où le modèle économique de beaucoup d'entreprises repose sur la publicité ciblée. Pour ces entreprises, la collecte massive de données est donc une priorité pour celles-ci. La vie privée et la sécurité passent en second.

D'autres entreprises travaillant sur l'Internet des objets (IdO) ont pour modèle économique la simple vente d'appareils électroniques et collectent des données en lien avec leur finalité première. Cependant, cela n'implique pas un comportement irréprochable et une capture excessive de données reste possible. Trouver quelles données sont les plus pertinentes à collecter et ne pas dépasser la quantité nécessaire au bon fonctionnement d'un service n'est pas une tâche aisée. Il n'est alors pas surprenant de voir des excès, même pour un modèle économique qui n'est pas basé sur la collecte massive de données.

Pourtant il est important aujourd'hui de considérer les aspects vie privée et sécurité dès la conception. En conséquence, des régulations et de la transparence sont nécessaires pour éviter de futurs scandales.

Depuis cette publication, l'Internet des objets n'a cessé d'évoluer positivement du point de vue du respect de la vie privée et du point de vue de la sécurité, particulièrement en Europe avec l'arrivée et la mise en application, en 2018, du RGPD. On constate de plus en plus que les fabricants considèrent la vie privée comme une composante prioritaire du développement, notamment lorsqu'on communiquent avec eux au sujet de nos données et de la sécurité de leur service. Les chartes de vie privée montrent à quel point certains fabricant prennent à coeur cette mission. On voit maintenant apparaître des alternatives aux grands acteurs. Ces alternatives mettent volontairement l'accent sur la vie privée, faisant presque de cette dernière un modèle économique à part entière. Des projets libres voient le jour, des initiatives sont prises pour redonner aux utilisateur le pouvoir sur leurs données et, du fait de beaucoup de scandales qui se sont produits ces dernières années, la vie privée est devenu un sujet de débat de haute importance.

1.2 Motivations

Plus le temps passe, plus des avancées sont faites en matière de vie privée et sécurité, mais certains défauts ne font que changer de forme. L'adoption massive d'un modèle économique basé sur la collecte massive de données et la publicité ciblée par les plus grands acteurs du numérique a une influence forte sur les façons d'innover et d'implémenter des idées.

L'incompréhension des utilisateurs devant une telle complexité demeure. Comprendre ce qu'est la vie privée et la sécurité est déjà une tâche complexe. Comment agir de la bonne manière pour reprendre le contrôle de sa vie privée en est une autre. L'intégration à cette complexité de nombreux protocoles et standards hétérogènes n'arrange en rien cette incompréhension. Entre Wi-Fi, Bluetooth, ZigBee, Z-Wave, Thread, IP, IPv6, 6LoWPAN, MQTT, COAP ou encore KNX, il y a de quoi être perdu. On ne comprend plus quel protocole sert à quoi, comment il s'utilise, avec quoi il est compatible, quels sont ses avantages et inconvénients, si ce sont des protocoles standardisés ou non. Pour un utilisateur, cette hétérogénéité peut être un frein conséquent dans l'adoption de nouveaux comportements. Il est facile de ne pas chercher à comprendre et renoncer à protéger ses données sous prétexte que les rouages de l'Internet des objets sont inaccessibles sans une certaine expertise. D'ailleurs, même pour un expert, la liste des connaissances à avoir est trop longue pour prétendre tout connaître. Avoir une vue globale de la situation actuelle est primordial pour décider des stratégies à adopter dans l'optique d'un futur où la technologie serait plus respectueuse de la vie privée et moins vulnérable aux attaques informatiques.

Aujourd'hui, la tendance est aux smartphones et aux objets connectés. La collecte massive de données se voit exacerbée dans un tel contexte. Tout prétexte est valable pour venir s'immiscer davantage dans la vie des utilisateurs afin de collecter toujours plus. On peut donc montrer que du fait de la nouveauté des objets, et en l'absence de transparence, de lignes directrices, de sensibilisation, de formations, de régulations, ou encore de sancitons, les mauvaises pratiques sont courantes.

Dès lors, il est crucial d'analyser la situation, de déterminer quelles sont les bonnes et mauvaises pratiques employées, leur étendue, leur fréquence et leur sévérité. Toutes les entreprises ne sont pas logées à la même enseigne lorsqu'il s'agit de concevoir un nouveau service, une nouvelle application ou un nouvel objet connecté, notamment vis-à-vis de la vie privée et de la sécurité. En effet, les petites entreprises peuvent rencontrer des difficultés que les grandes ne rencontreront pas. Aussi est-il important de découvrir ces disparités.

Corriger les problèmes rencontrés devient alors plus facile et il reste du ressort de la CNIL et de ses équivalents européens de prendre acte de ces observations. Depuis la mise en application du RGPD, la réglementation s'est renforcée et permettra probablement de pénaliser avec plus de justesse les mauvaises pratiques, favorisant ainsi des stratégies plus propres se basant notamment sur la considération des questions de vie privée dès la conception d'un produit. Du reste, nous ne sommes pas à l'abri de nouvelles découvertes telles que des techniques dont certains acteurs pourraient faire usage pour soutirer furtivement certaines informations auxquelles ils ne devraient pas avoir accès. Les travaux présentés dans cette thèse devront par conséquent être répétés pour s'adapter aux nouvelles tendances du domaine.

1.3 Contributions et organisation de la thèse

Dans cette thèse je présenterai le domaine particulier des maisons intelligentes, et montrerai que les pratiques vertueuses auxquelles ont pourrait s'attendre de la part des grands acteurs en particulier ne sont toujours pas mises suffisamment en pratique.

Le chapitre 2 dresse un état de l'art de l'Internet des objets, ses protocoles, ses usages et ses problèmes de vie privée et de sécurité.

Le chapitre 3 concerne l'étude des problèmes de vie privée et de souveraineté concernant l'écosystème des ampoule connectées et de leurs différents moyens de contrôle. Cet écosystème est choisi volontairement pour sa simplicité, et a permis de prouver que même avec une telle simplicité, les conséquences sur la vie privée ne sont pas anodines. Nous verrons que les problèmes de vie privée et de sécurité pourtant simples et connus sont toujours d'actualité, et que même si la situation s'est améliorée sur certains aspects, elle stagne dans d'autres.

Nous constaterons les diverses erreurs commises par différents acteurs concernant un seul type d'objet (les ampoules) dans le but de fournir une comparaison pertinente.

Le point de vue adopté dans cette thèse est original en se concentrant uniquement sur la partie contrôle des ampoules, c'est-à-dire les différentes applications et appareils qui peuvent être utilisés pour faire fonctionner une même ampoule, ce qui, à ma connaissance, n'a pas été fait dans la littérature scientifique. En effet, il est important d'observer le comportement d'un objet et les problèmes associés lorsqu'on utilise son application officielle, mais il est aussi important d'observer les différences qui apparaissent lorsqu'on n'utilise plus l'application officielle. Différents moyens de contrôle (smartphone, applications officielles, applications tierces ou encore objets connectés) optent pour différentes techniques de contrôle (accès local, distant, automatisation d'actions, transmissions chiffrées ou encore transmission en clair), ce qui engendre des comportements très divers.

Je mettrai dans un premier temps en lumière les problèmes de vie privée, de souveraineté, d'hégémonie des grands acteurs, de manque de transparence, de manque de sécurité et de mécanismes de protections contre les fuites de données et de collecte abusive de données à caractère personnel.

Le chapitre 4 décrit le problème du consentement dans les configurations des applications, et de l'obtention d'un consentement libre, spécifique, éclairé et univoque. Toutes ces observations mènent à un problème global : il est presque impossible pour des utilisateurs de comprendre les rouage de ces technologies, et dans cette incompréhension, leur seul choix est de faire une confiance aveugle aux entreprises qui leur fournissent des services, applications et objets connectés. Malheureusement, au vu des observations faites dans cette thèse, une telle confiance serait accordée à tort, et il est important de corriger cela.

Pour pallier ce problème, je présenterai des pistes simples pour atténuer les problèmes rencontrés lors de notre étude, afin de redonner à l'utilisateur une partie du contrôle qui lui est dû.

Je terminerai par une conclusion et des perspectives sur la situation actuelle, les tendances, et les recommandations possibles afin d'améliorer la situation.

1.4 Publications

M. Thiery, V. Roca, A. Legout, "The Chaotic Travel of Smart-Bulb Data : the Privacy Implications of Control Techniques", *en cours de soumission.*

M. Thiery, V. Roca, A. Legout, "Privacy implications of switching ON a light bulb in the IoT world", pré-publication sur HAL, 2019. https://hal.inria.fr/hal-02196544

L. Demir, M. Thiery, V. Roca, J-L. Roch, J-M. Tenkes, "Optimizing dm-crypt for XTS-AES : Getting the Best of Atmel Cryptographic Co-Processors", SECRYPT 2020, 17th International Conference on Security and Cryptography, Jul 2020, Paris, France. Version longue disponible sur https://hal.archives-ouvertes.fr/hal-02555457.

L. Demir, M. Thiery, V. Roca, J-L. Roch, J-M. Tenkes, "Improving dm-crypt performance for XTS-AES mode through extended requests : first results", Grehack 2016, The 4th International Symposium on Research in Grey-Hat Hacking - aka GreHack , Nov 2016, Grenoble, France. https://hal.inria.fr/hal-01399967

Chapitre 2

État de l'art

Dans ce chapitre, j'aborderai les différents axes de recherche actuels concernant la vie privée et la sécurité dans l'Internet des objets.

J'y ferai une mise en contexte afin d'avoir une bonne idée de la complexité de cet écosystème. Je montrerai les différents objets et usages existants et me focaliserai particulièrement sur le domaine de la maison intelligente, dont je donnerai une définition particulière. J'expliquerai aussi qui sont les acteurs dans ce domaine et quels sont leurs modèles économiques.

Je montrerai l'étendue des protocoles existants sur le modèle OSI, leur complexité et leur hétérogénéité, leurs objectifs et caractéristiques particulières.

Je présenterai ensuite la littérature scientifique au sujet de la vie privée dans le contexte particulier de l'Internet des objets. J'expliquerai comment on capture des données, l'usage classique et l'usage non conventionnel qu'on peut en faire, comment on peut analyser ces données, quelles sont les défenses pour empêcher une personne malicieuse de faire ce genre d'analyse, le consentement et quelques autres préoccupations que l'Internet des objets a créées.

Je continuerai par la littérature scientifique au sujet de la sécurité, toujours concernant l'Internet des objets. Je listerai les types de vulnérabilité connues, dont les injections de commandes sur les AVP, et parlerai des défenses associées dès que cela est possible.

Je conclurai sur le positionnement de cette thèse vis-à-vis de l'état de l'art actuel, à savoir : pourquoi j'ai choisi de me concentrer sur les différents moyens de contrôle des objets plutôt que sur les objets eux-même.

2.1 Objets connectés

Cette section s'intéresse aux objets connectés dans l'Internet des objets et en particulier dans les maisons intelligentes.

Je présenterai mon point de vue sur ce que sont les maisons intelligentes, sur ce qui peut s'y trouver et sur les acteurs qui y participent, ainsi que les choix que ceux-ci font en fonction de leurs modèles économiques.

2.1.1 Contexte et définitions

L'IdO couvre une grande quantité de domaines dont, par exemple, les villes, la santé, les habitats, l'industrie, l'énergie, l'agriculture ou encore la sureté. Des termes ou expressions voient le jour régulièrement pour parler d'un contexte particulier. On parle souvent de villes intelligentes, de maisons intelligentes, de mesure de soi (le domaine de l'IdO dédié à la

mesure des caractèristiques physique ou de santé de l'utilisateur), et toutes ces expressions décrivent des contexte aux frontières souvent floues.

Pour illustrer ce flou, prenons l'exemple des maisons intelligentes sur lesquelles je me suis focalisé dans cette thèse. Définir ce qu'est une maison intelligente ou connectée est complexe. Est-ce qu'on considère que c'est une maison qui contient des objets connectés ? On serait tenté de dire que oui, mais on voit bien qu'avoir juste une bouilloire connectée ne rend pas vraiment ma maison connectée ou intelligente. Ma bouilloire est-elle vraiment connectée d'ailleurs ? Certaines définitions diraient qu'une connexion à Internet est requise pour avoir une maison connectée, pourtant si ma maison fonctionne purement localement et intègre une multitude d'objets qui communiquent entre eux sans Internet, elle est alors plus connectée et intelligente qu'une maison avec une seule bouilloire connectée à Internet. Est-ce qu'on suppose que seul un objet connecté destiné à l'usage domestique peut rendre une maison connectée ? Si on considère une balance connectée et une montre intelligente, on peut voir que les deux sont des outils de mesure de soi, ce qui est un domaine différent de la maison intelligente. On voit bien ici qu'il y a pourtant une intersection entre ces deux domaines. Pourtant la montre semble plus volatile alors que la balance est bien ancrée à la maison. Mais une montre intelligente peut être une porte d'entrée vers une grande quantité d'autres objets connectés, ce qui en fait, comme le smartphone, un objet connecté particulier, une interface. Encore une chose : quelle est la différence entre domotique et maison connectée ? Est-ce que ce sont des synonymes ? ou est-ce qu'une maison connectée impose l'usage de protocoles différents, ou la notion de communication plus accrue entre les appareils, ou encore une facilité d'installation ?

Voici quelques règles qui pourraient définir ce que sont ces notions selon moi :

1. Une maison peut être plus ou moins connectée, allant d'un seul objet connecté à une multitude. Ce n'est pas du tout ou rien mais bien un spectre.
2. Une maison devient connectée dès lors qu'un objet connecté y est présent et à la condition que cet objet ait été créé dans l'intention première d'être utilisé à la maison.
3. Une maison est « intelligente » à partir du moment où elle est capable d'une certaine autonomie. Exemples : démarrer la bouilloire à 18h, fermer les volets lorsque la nuit tombe, ou régler la température intérieure en fonction de la présence d'habitants. Une ampoule connectée, à elle seule, ne rend pas une maison « intelligente ».

2.1.2 Panorama des objets connectés

En se limitant à ces définitions, les objets connectés des maisons intelligentes et/ou connectées sont divisés en plusieurs catégories. La figure 2.1 illustre ces catégories ainsi que quelques exemples notoires d'objets connectés qu'elles renferment. Les catégories choisies ici l'ont été parce qu'elles sont suffisamment abstraites pour englober une portion représentative des objets connectés dans les maisons intelligentes. Évidemment, d'autres catégories peuvent être définies.

Faire une liste exhaustive de ces objets est impossible tant il en sort de nouveaux fréquemment. On peut cependant constater que certains acteurs sont plus populaires et ont fait de l'IdO une branche à part entière de leurs produits comme Google (qui a phagocyté Nest), Amazon, Apple, Philips, Samsung, Honeywell ou LIFX. Quelques nouveaux arrivants comme IKEA semblent tracer leur chemin vers cette liste également.

La figure montre également la grande complexité des champs d'application de l'IdO dans les maisons intelligentes. On constate que beaucoup de domaines existent et que beaucoup d'objets se trouvent à l'intersection de plusieurs domaines. Ces objets doivent donc répondre à des enjeux multiples pouvant demander des expertises très variées.

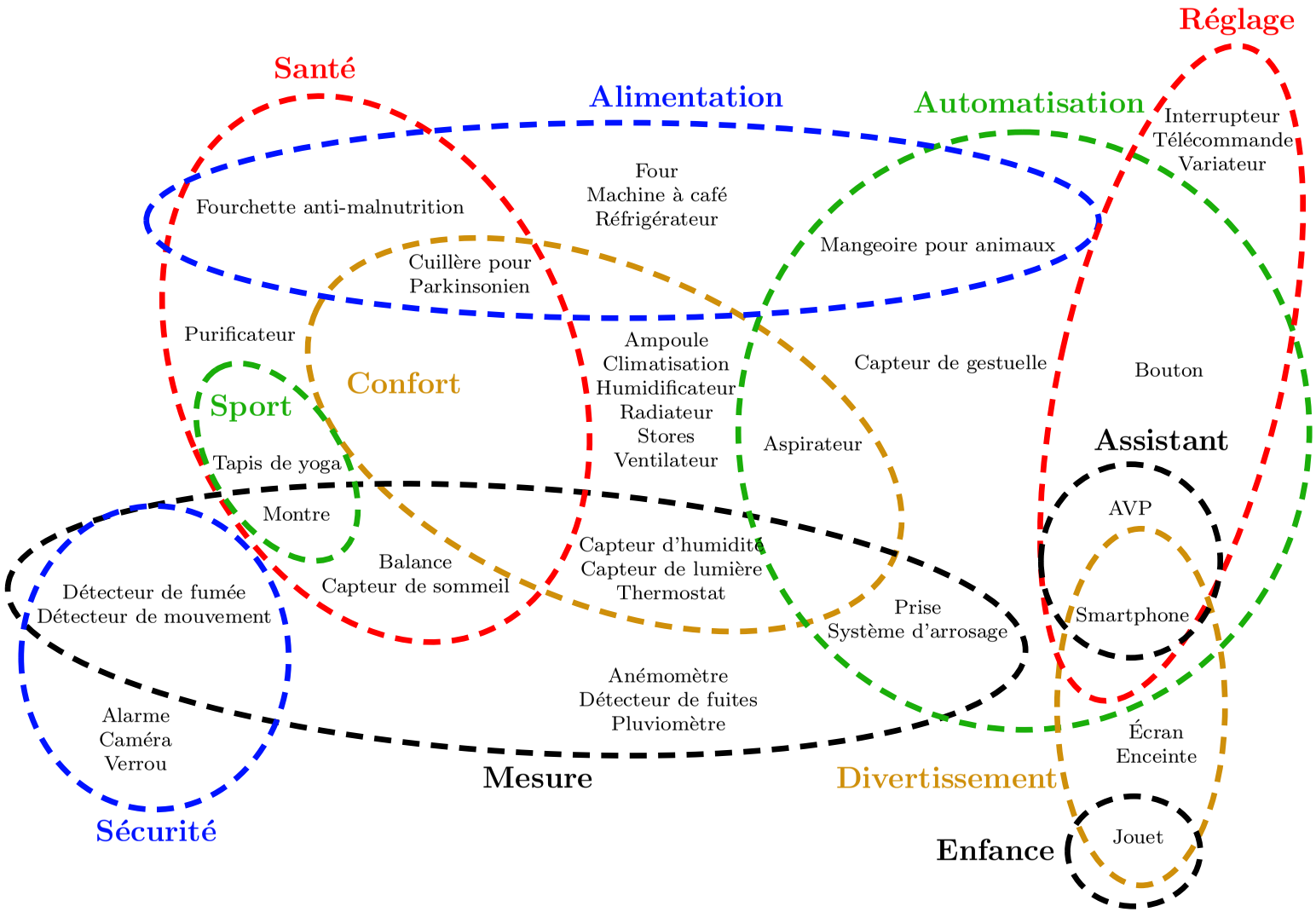
Les contraintes en matière de vie privée sont aussi totalement différentes d'un objet à l'autre. Pris à part, un pluviomètre n'apporte a priori pas un risque significatif sur la vie privée, tandis qu'un smartphone a un impact nuisible potentiel gigantesque. En revanche, indirectement, un pluviomètre peut poser plus de problème si, par exemple, il est un noeud de transit dans un réseau maillé. La simple position d'un objet anodin peut aussi le rendre dangereux pour le respect de la vie privée. Par exemple un capteur de lumière extérieur peut s'avérer moins dangereux qu'un capteur d'intérieur, puisque l'un mesure principalement la luminosité solaire, tandis que l'autre est capable de mesurer la luminosité solaire et artificielle dans laquelle les habitants baignent. Cela veut dire qu'une variation soudaine de luminosité dans une salle peut être la conséquence de la présence d'une personne, et devient donc une source de problème de vie privée.

La même observation est valable pour les contraintes en matière de sécurité, que ce soit d'un point de vue informatique ou physique. Un verrou connecté, par exemple, doit à tout prix être robuste et sans failles car une attaque réalisée sur ce genre de produit pourrait permettre à un cambrioleur de pénétrer dans la maison qui aurait dû être protégée. À l'inverse, une attaque informatique sur un tapis de yoga n'aura pas de grand impact sur la sécurité. Enfin, une attaque informatique réalisée sur un four connecté peut mener à de graves conséquences physiques sur les habitants, alors qu'une montre subissant ce genre d'attaque n'aura pas d'effet physique nuisible direct.

Un dernier point similaire : chaque objet a son lot de considérations éthiques. Un mangeoire pour animaux ne soulève pas de problème éthique lié à l'homme, mais un jouet pour enfant en soulève beaucoup, d'autant plus que cet objet concerne quelqu'un qui n'a pas la même capacité de consentement qu'un adulte responsable. D'ailleurs, la nécessité pour cet objet d'être configuré par un adulte pose des questions éthiques qui se situent à l'intersection des droits de l'enfant et de l'adulte.

Certains domaines comme la santé, la sécurité, l'enfance ou les assistants sont par essence des domaines où toutes ces questions de vie privée, sécurité et éthique doivent être posées. Mais il est aussi très facile de tomber dans le piège de la confiance envers les objets faisant partie d'autres domaines moins sensibles sous prétexte qu'ils ne semblent pas présenter de risques. On constate souvent, comme nous le verrons dans le chapitre 3, que l'utilisation d'objets aussi anodins soient-ils, peuvent causer des conséquences inattendues.

Figure 2.1: *Catégories et exemples d'objets connectés dans les maisons intelligentes. Ce graphique illustre la complexité et l'hétérogénéité des écosystèmes de l'IdO.*



2.1.3 Acteurs et modèle économique

Bien qu'on puisse citer des exemples d'objets dont le respect de la vie privée et la sécurité peuvent poser question du fait, par exemple, de leur sensibilité (comme un médicament contraceptif [22]) ou leur provenance (comme un objet d'appels vidéo développé par Facebook [23]), on peut également nommer des objets ou services moins connues mais a priori plus respectueux de la vie privée comme Mycroft [24] et Snips [25] récemment racheté par Sonos [26], pour les assistants personnels par exemple. Ces derniers exemples axent très clairement leur communication sur le respect de la vie privée avant tout le reste, et montrent les techniques employées pour atteindre ce but. Pour les deux exemples donnés ici, il s'agit d'enregistrer et analyser la voix de l'utilisateur localement. Aucune interaction avec l'extérieur n'est nécessaire à cette étape, ce qui en fait une solution respectueuse dès la conception.

Néanmoins, il est clair que le mot d'ordre pour une bonne partie des fabricants d'objets connectés, est la capture de données, massive dans certains cas, à des fins de profilage utilisateur et *in fine* de publicité ciblée.

C'est en effet un modèle économique que l'on observe dans nos travaux, notamment adopté par les fabricants d'AVP. Plus ils supportent d'objets, plus le nombre de données et les croisements et inférences possibles sur ces données augmentent, et donc plus le ciblage peut-être pointilleux. Nous avons également constaté que certains fabricants d'objets connectés cherchent à garder un lien permanent avec leurs objets, quitte à les déposséder de leur autonomie, afin de collecter les données qu'ils capturent. C'est un problème dont on discutera dans le chapitre 3, où l'on verra par exemple que l'allumage d'une ampoule, dans certaines situations, passe nécessairement par une connectivité à Internet.

Pour les fabriquants de smartphone et les concepteurs de systèmes d'exploitation mobile, c'est aussi une aubaine puisque c'est grâce à leur systèmes et aux applications qu'ils embarquent que la plupart des interactions avec les objets se feront, ne serait-ce que pour leur configuration. Toutes les données passent par leur système, ce qui laisse le champs libre à des incartades discrètes délétères pour l'utilisateur, même si un croisement des données est souvent nécessaire.

Les enceintes et les robots connectés sont de nouveaux vecteurs de divulgation également, au même titre que les smartphone. Ils permettent d'attaquer un nouveau milieu jusqu'alors inexploré : la maison. Maintenant, ces objets peuvent capturer des données non seulement sur l'utilisateur de l'objet, ce qui n'a plus vraiment de sens dans ce contexte, mais sur *les utilisateurs* de l'objet : la famille. Contrairement au smartphone, ces objets ont une place moins discrète (ils ne sont pas confinés dans une poche mais restent à la vue de tous) et plus discrète (ils s'intègrent au décors domestique, font partie de la famille sans qu'on ne leur prête une réelle attention). Ce placement, idéale pour l'écoute intrusive, permet aux enceintes et robots connectés, de capturer une masse de données beaucoup plus conséquente qu'un smartphone, et atteint des acteurs au consentement ambigu comme les enfants, les voisins, les invités, etc..

Les développeurs sont également des acteurs incontournables de ce modèle. Ils peuvent à leur gré ou malgré eux (via des parties tierces ou librairies au comportement douteux) permettre la collecte de plus de données que nécessaire. Leur principal revenu est la publicité. Pour obtenir une publicité à plus grand revenu, il faut des données plus précises, ce que le développeur peu scrupuleux sera tenté de capturer. C'est ensuite un cercle vicieux qui démarre entre tous ces acteurs : plus de données donnent un ciblage meilleur, qui donne un revenu plus grand.

On remarque un paradoxe nouveau qui survient avec l'apparition des objets connectés. Jusqu'à maintenant l'idée de ce modèle économique était celui dont découle l'adage « *si c'est gratuit, c'est vous le produit* », à savoir que la publicité permettait de payer le produit à votre place, mais que celle-ci se servait de vos données pour se rémunérer. Pourtant,

les nouveaux produits s'inscrivent dans une démarche différente. Premièrement, on achète l'objet connecté contrairement aux applications (qu'on doit toujours télécharger d'ailleurs). Deuxièmement, l'objet collecte des données et les envoie parfois à leur fabriquant. Et troisièmement, la publicité n'est plus nécessairement affichée ni diffusée sur le produit, mais peut être affichée ailleurs, par exemple envoyée par courriel, affichée sur une application (liée ou non à l'appareil), sur des réseaux sociaux liés à l'appareil, etc.. Ce qui veut dire qu'au final nous payons pour un objet et/ou un service qui, de toute façon, capture des données sur nous et s'en sert pour gagner de l'argent en plus. Pour illustrer ce propos, on peut citer le robot aspirateur de Roomba [13] que l'utilisateur doit acheter au prix fort, et qui a la capacité de créer une carte virtuelle de sa maison de laquelle Roomba peut extraire des informations (acoustique de la pièce, mobilier manquant, nombre de pièces, surface, nombre d'habitants estimé) pour ensuite les revendre au besoin. Ce type de données pourrait leur permettre, s'ils le souhaitaient, d'estimer le niveau de vie des habitants.

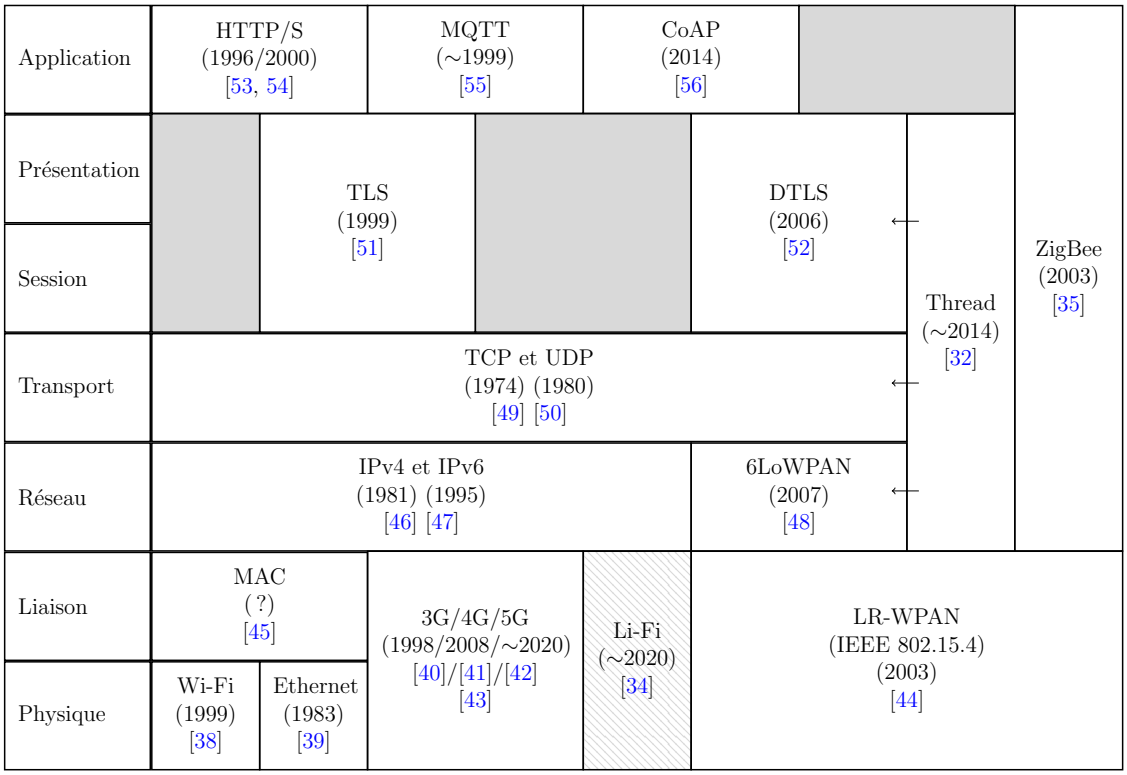
2.2 Protocoles et connectivité

Cette section s'intéresse aux protocoles qui constituent l'Internet des objets, à leur complexité, leur hétérogénéité, leurs objectifs, leurs caractéristiques, leurs avantages et leurs inconvénients.

En 2015, Derhamy et al. [27] ont cherché à donner un panorama de l'existant, en listant 17 frameworks et plateformes commerciales de l'époque utilisés pour développer les applications de l'Internet des objets (IdO). Ils analysent le tout sous l'angle de la sécurité et vie privée et dégagent des tendances. Cependant cet article n'est plus représentatif de la situation actuelle et parfois incomplet, notamment par rapport aux protocoles existants. En 2017, Marksteiner et al. [28] listent les protocoles dédiés aux maisons intelligentes, qui correspondent en partie à notre sélection des plus importants d'entre eux (voir section 2.2.1), mais qui encore une fois reste incomplet. Plus ancien mais assez révélateur, Chan et al. [29] abordent en 2012 le statut particulier de la mesure de soi et en énonce ses défis. C'est un sous-domaine auquel nous nous intéresserons assez peu mais qui représente une part importante du marché et n'est donc pas négligeable. Ces trois articles permettent déjà d'avoir un premier aperçu de l'Internet des objets, son développement, ses protocoles, et ses différents domaines, entre autres dans le cadre des maisons intelligentes et de la mesure de soi, mais on peut également se référer à un article de 2017 dans lequel Saha et. al [30] énumèrent les tendances récentes, notamment en matière de protocoles dans l'Internet des Objets (IdO) et mentionnent par exemple WiFi-direct [31], Thread [32], RFID [33], Li-Fi [34], ZigBee [35] et Z-Wave [36], qui font aussi partie de notre sélection, mais la liste fournie par cet article reste incomplète. Certaines de ces technologies sont d'ailleurs trop récentes, comme le Li-Fi, pour être significativement répandues (voir section 2.2.1) dans les objets connectés. Bien que celui-ci ait toute sa place dans notre inventaire, il est important d'observer que son statut réel est encore spéculatif. On peut tout de même retenir Wi-Fi, Bluetooth, ZigBee et Z-Wave que l'on retrouve sur beaucoup de produits et qui, pour les deux premiers, sont particulièrement répandus du fait des capacités de connectivité des smartphones. Les smartphones ayant un rôle central dans l'IdO, il n'est pas surprenant que Bluetooth et Wi-Fi soit deux protocoles centraux eux aussi.

Mais au delà des protocoles en tant que tels, c'est leur complexité et leur hétérogénéité qui est révélatrice de la confusion dans laquelle se trouve l'IdO actuellement.

Figure 2.2: *Étendue de divers protocoles sur les couches du modèle OSI. Les protocoles listés sont capables de communiquer directement sur Internet, sans procéder à un changement de protocole. Entre parenthèses se trouvent les dates des premières spécifications des protocoles. Un symbole environ est ajouté quand aucune date fiable n'a été trouvée. Li-Fi est hachuré pour indiquer que son placement est une suggestion, son usage futur n'étant pas encore établi concrètement. Les flèches indiquent que le protocole Thread implémente 6LoWPAN, UDP et DTLS. ZigBee a également développé un équivalent d'IP (ZigBee IP), mais ce dernier est très peu adopté et n'est donc pas montré ici.*

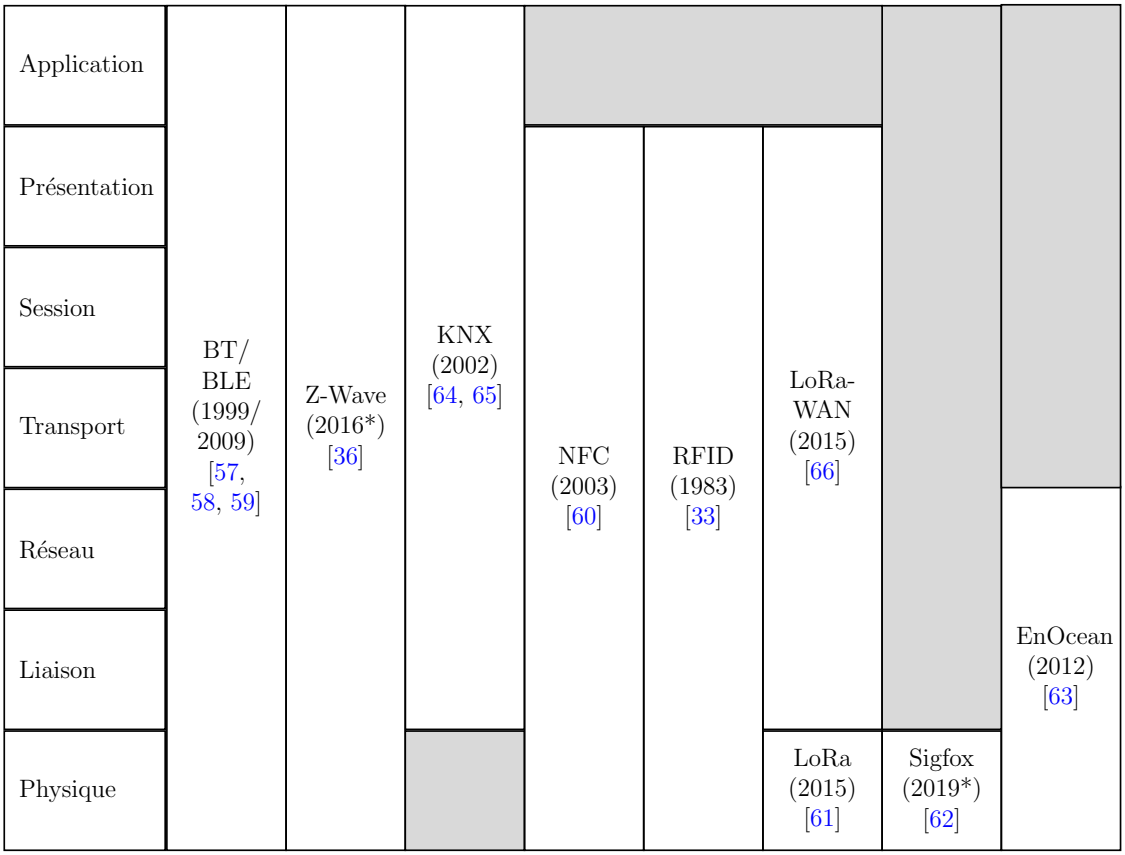


2.2.1 Modèle OSI

Pour comprendre la confusion que peut amener la myriade de protocoles dont fait usage l'Internet des Objets (IdO), les tableaux 2.2 et 2.3 nous montrent l'étendue des principaux protocoles sur les différentes couches du modèle OSI [37].

On constate plusieurs choses. D'abord, certains protocoles sont des alternatives l'un de l'autre, comme COAP et MQTT. D'autres sont des alternatives mais dont l'étendue est différentes comme ZigBee et Thread qui utilisent tous les deux la même base IEEE 802.15.4, mais où ZigBee s'étend jusqu'à la couche application. Thread implémente également les protocoles DTLS, UDP et 6LoWPAN (voir figure 2.2). On constate aussi que certains protocoles gèrent toutes les couches, comme Bluetooth, Z-Wave, etc., tandis que d'autres n'implémentent que la couche physique comme SigFox, ou encore, en étendant à la couche de liaison, comme Li-Fi, Wi-Fi et 3G/4G/5G, ou jusqu'à la couche présentation comme LoRaWAN/LoRa. KNX est aussi à part puisqu'il est capable de supporter plusieurs couches physiques, dont Ethernet.

Figure 2.3: *Étendue de divers protocoles sur les couches du modèle OSI (partie 2). Cette fois, les protocoles listés sont destinés à une utilisation locale. Ils devront donc procéder à un changement de protocole pour pouvoir communiquer sur Internet. KNX est particulier car il supporte plusieurs couches physiques, dont Ethernet. Les protocoles sont ici calqués sur le modèle OSI, bien que ce modèle ne soit pas réellement adapté à ces protocoles.*



2.2.2 Caractéristiques

Description des protocoles

Pour compléter cette vue du modèle OSI, le tableau 2.1 (respectivement 2.2) liste les caractéristiques des protocoles couvrant les couches physique à réseau (respectivement transport à application) du modèle OSI nommés précédemment. Faire une comparaison exhaustive est impossible mais on peut déjà dégager quelques tendances et différences majeures entre les protocoles.

Beaucoup de ces protocoles sont déjà relativement anciens et bien connus aujourd'hui, mais il est nécessaire de se focaliser un certain temps sur quelques protocoles récents et très présents dans l'Internet des objets aujourd'hui.

6LoWPAN (IPv6 Low power Wireless Personal Area Networks) est un protocole destiné à transmettre des paquets IPv6 sur le protocole IEEE 802.15.4 qui concerne les LR WPAN (Low Rate Wireless Personal Area Network). Il est le résultat d'un groupe de travail sur la problématique de l'intégration du protocole IP dans les communications d'appareils à ressources contraintes. Le but de ce protocole est donc de fragmenter les paquets IPv6 et de compresser leurs entêtes pour qu'ils puissent être envoyés sur le protocole IEEE 802.15.4.

Thread est un protocole en réseau maillé utilisant 6LoWPAN, ce qui lui permet de se comporter comme un noeud normal dans un réseau local grâce au protocole IP. Tout protocole basé sur IP peut donc être utilisé par Thread. Il a été conçu pour qu'un utilisateur puisse utiliser, entre autres, son smartphone afin de connecter simplement un objet au réseau. Dans ce genre de réseau, certains noeuds vont prendre le rôle de routeur afin de relayer les paquets jusqu'à internet grâce à une interface Wi-Fi ou Ethernet.

ZigBee est un protocole en réseau maillé basé sur IEEE 802.15.4 similaire à Thread, à ceci prêt qu'il n'utilise pas 6LoWPAN et, par conséquent, ne peut pas utiliser IP, bien que certains travaux ont été faits dans ce sens. D'autres travaux sont également en cours dans le but de pouvoir interconnecter ZigBee et Thread. L'absence d'IP fait que ZigBee implémente ses propres mécanismes de routage. ZigBee définit une couche application supplémentaire par rapport à Thread qui peut être avantageuse lors de l'utilisation de systèmes d'éclairage mais désavantageuse dans d'autres cas.

MQTT (Message Queuing Telemetry Transport) est un protocole basé sur TCP/IP destiné à la transmission de paquets de courte taille. Il fonctionne sur un système d'abonnement et publication. Il est nécessaire pour cela de dédier un appareil au routage des paquets entre plusieurs clients, appareil que MQTT appelle le « broker ». Cela permet ainsi de retenir certains messages jusqu'à ce qu'un client destinataire s'abonne au broker. MQTT ne fournit aucune sécurité mais peut utiliser TLS dans ce but.

COAP (Constrained Application Protocol) est un protocole basé sur l'architecture REST et le modèle client-serveur et dédié aux appareils contraints. Il a certaines similarités avec HTTP du fait de son architecture REST, mais en diffère de part son utilisation d'UDP en lieu et place de TCP. Comme MQTT, COAP ne fournit pas de sécurité mais peut utiliser DTLS, l'équivalent de TLS pour UDP. COAP est conçu pour être compatible avec 6LoWPAN.

BLE (Bluetooth Low Energy) est une version basse consommation de Bluetooth intégrée à la version 4.0 de ce dernier bien qu'il soit incompatible avec le mode standard de Bluetooth. Cette incompatibilité est due au fait que BLE est destiné aux paquets de petite taille. Tout comme Bluetooth en revanche, BLE permet uniquement la transmission de paquets locaux et n'est pas destiné à la transmission de paquets sur Internet.

Table 2.1: *Liste des protocoles des couches physique à réseau et de leurs caractéristiques.*

Protocole	Encapsulation/ Standard	Topologie	Portée	Débit	# Noeuds	Basse consommation	Chiffrement
3G/4G/5G	—	—	0.3 à 16 km	<150 Mb/s	—	—	KASUMI/SNOW 1.0/2.0/3G
6LoWPAN	IEEE 802.15.4	étoile ou maillée	—	<250 kb/s	—	oui	AES-CCM
BT/BLE	IEEE 802.15.1	étoile ou scatternet	10 à 200m	1 à 3 Mb/s	32 767	non/oui	E0/SAFER+/AES-CCM
EnOcean	ISO/CEI 14543-3-10	maillée	30 à 300m	125 kb/s	—	oui	AES-CBC/VAES
Ethernet	IEEE 802.3	étoile	—	10 Gb/s	—	non	non
IP	MAC	—	—	—	—	non	non
KNX	Ethernet et autres,	libre	—	9 Mb/s	—	—	—
Li-Fi	IEEE 802.15.7	—	<5 m	10 à 40 Mb/s	—	—	—
LoRaWAN	LoRa	étoile	5 à 15 km	0.3 à 37.5 kb/s	—	oui	AES-CTR
NFC	ISO/CEI 14443	—	10 cm	424 kb/s	—	—	non
RFID	ISO/CEI 14443	—	10 cm	424 kb/s	—	—	non
SigFox	SigFox	étoile	10 à 50 km	0.1 à 0.6 kb/s	—	oui	non
Thread	IEEE 802.15.4	maillée	20 à 30 m	250 kb/s	250	oui	AES-CCM
Wi-Fi	802.11	étoile	70 m	600 Mb/s	—	non	WPA2/CCMP/AES
Z-Wave	Z-Wave	maillée	30 m	10 à 100 kb/s	232	oui	AES
ZigBee	IEEE 802.15.4	maillée	10 m	40 à 250 kb/s	65000	oui	AES

Table 2.2: *Liste des protocoles des couches transport à application et de leurs caractéristiques. * = protocoles supportant un grand nombre de types de chiffrements, dont AES.*

Protocole	Encapsulation	Chiffrement
COAP	UDP ou DTLS	non ou DTLS
DTLS	UDP	oui*
HTTP/S	TCP ou TLS	non ou TLS
MQTT	TCP ou TLS	non ou TLS
TCP	IP ou 6LoWPAN	non
TLS	TCP	oui*
UDP	IP ou 6LoWPAN	non

Les défis relevés par les protocoles

L'un des premiers défis de l'IdO concerne la consommation d'énergie. L'augmentation du nombre d'objets connectés a pour effet inévitable d'augmenter la consommation énergétique. Dès lors, le but de certains protocoles est de minimiser cet impact énergétique. Ce défi est donc relevé par 6LoWPAN et les protocoles basés sur lui comme Thread ou compatibles avec IEEE 802.15.4 comme ZigBee. Il est également relevé, dès la couche physique, par BLE, la version basse consommation de Bluetooth, LoRaWAN/LoRa, SigFox, EnOcean et Z-Wave. Bien que certains protocoles plus applicatifs soient indiqués comme ayant une basse consommation, cela est dû aux protocoles sous-jacents qui le sont. Les protocoles basse consommation partent du principe que les petits appareils faibles en ressources et peu consommateurs d'énergie doivent être en mesure de participer à l'IdO. Pour ce faire, ces protocoles emploient des mécanismes de compression [48], réduisent leur portée et leur bande passante [35, 44], supportent des objets dont les composants consomment peu [56], voire récoltent de l'énergie [63] (mouvements, lumière, etc.), ou encore implémentent un mode dormant [58]. Toutes ses solutions permettent de réduire tant que possible la consommation énergétique de l'appareil. À titre d'exemple, Pour un objet consommant 1 W en Bluetooth, son homologue basse consommation Bluetooth Low Energy (BLE) consomme entre 0,5 et 0,01 W [58], soit une division par 2 dans le pire des cas, et par 100 dans le meilleur. D'un autre côté, certains protocoles comme le Wi-Fi n'ont pas cet objectif d'économie d'énergie, en partie du fait de leur ancienneté, puisque cette problématique n'était pas autant d'actualité à leur création.

Deuxième défi : la sécurité. Cette sécurité implique beaucoup de fonctionnalités à remplir comme le chiffrement, l'authentification, des preuves d'authenticité, etc.. Cependant, le chiffrement est une nécessité absolue pour qu'une communication soit sécurisée et est un bon élément de comparaison dans notre cas. On voit que la plupart des protocoles de la couches application reposent sur l'utilisation d'un protocole sous-jacent qui, lui, est chiffré ou non. Par exemple COAP et MQTT peuvent utiliser TLS (DTLS étant la version UDP de TLS). Les protocoles de couche physique comme Bluetooth, 3G/4G, LoRaWAN/Lora, Wi-Fi et Z-Wave utilisent leur propre chiffrement, ce qui permet théoriquement de protéger tous les protocoles encapsulés sur le segment, mais pas de bout en bout. Enfin ZigBee et Thread implémentent leur propre couche de chiffrement indépendamment de la couche physique, ce qui permet de se détacher, théoriquement de la couche physique. Pour plus de détails sur la sécurité, les figures 2.1 et 2.2 montrent quels types de chiffrements sont utilisés pour chaque protocole. On constate que la majorité des protocoles utilisent AES, qui est reconnu comme un type de chiffrement très robuste. Cependant, on se limite ici au chiffrement utilisé lors de la transmission de données. L'échange de clés, l'authentification, les preuves d'intégrités tout aussi essentiels, ne seront pas traités dans cette thèse.

Troisième défi : l'accessibilité. Une topologie en étoile est appréciée dans la plupart des réseau domestiques, le principale protocole utilisé dans un tel contexte étant le Wi-Fi, cependant dans le monde de l'IdO, elle n'est pas idéale. Partant du principe que la consommation énergétique doit rester la plus basse possible pour la plupart des protocoles de l'IdO, comme vu précédemment, un compromis doit être fait en matière de débit et/ou de portée. Au contraire, les réseau type Wi-Fi sont, du fait de leur topologie en étoile, contraints d'avoir un routeur central qui se chargera de relayer toutes les communications du réseau, ce qui oblige à avoir une portée suffisante pour atteindre tous les objets de la maison, et par conséquent, demande une consommation électrique relativement élevée. Pour pallier ce problème, et partant du principe que les objets connectés sont omniprésents dans une maison, les topologies maillées sont adaptées, puisqu'elles permettent d'avoir des objets prenant le rôle de relais, idéalement placés à courte distance d'autres objets. Ainsi, deux objets positionnés aux extrêmes opposés d'une maison peuvent communiquer entre eux en utilisant des objets intermédiaires comme relais. Ce mécanisme nécessite une courte distance entre chaque objet intermédiaire, et permet donc de conserver un débit correct sans impacter la consommation d'énergie (se référer au tableau 2.1). De telles topologies sont employées par les protocoles ZigBee, Thread et Z-Wave. Dans une certaine mesure, Bluetooth peut aussi être inclus dans cette catégorie grâce à sa topologie scatternet qui est très proche d'un réseau maillé à ceci pret qu'elle nécessite des noeuds maîtres et esclaves, ce qui n'est pas le cas des réseaux maillés purs. Cependant nous n'avons pas rencontré cette topologie particulière de Bluetooth lors de nos travaux, topologie qui reste encore relativement rare. Cette stratégie des réseaux maillés est efficace pourvu qu'il y ait une densité d'objets suffisamment élevée dans la maison pour que tout objet puisse atteindre n'importe quel autre objet.

Quatrième défi : la standardisation. Il existe plusieurs standards, dont principalement ceux listés dans le tableau 2.1 reproduits et expliqués ici :

IEEE 802.3 [67] concerne les réseaux câblés dont Ethernet ;

IEEE 802.11 [68] concerne les WLAN (Wireless Local Area Network) ou réseaux sans-fil locaux dont principalement le Wi-Fi ;

IEEE 802.15.1 [69] concerne les WPAN (Wireless Personal Area Network) ou réseaux sans-fil personnels et est dédié à Bluetooth ;

IEEE 802.15.4 [44, 69] concerne les LR WPAN (Low Rate WPAN) ou réseaux sans-fil personnels à bas débit comme ZigBee, Thread et 6LoWPAN ;

IEEE 802.15.7 [69] concerne les réseaux optiques comme Li-Fi ;

ISO/IEC 14443 [70] concerne les carte à circuits intégrés sans contact comme NFC et RFID ;

ISO/IEC 14543-3-10 [71] concerne les composants optimisés pour la récolte d'énergie.

Standards IETF (RFC) Une grande quantité de protocoles utilisés par l'Internet des objets sont standardisés par l'IETF. On peut citer IP (v4 et v6), 6LoWPAN TCP, UDP, TLS, HTTP ou encore COAP.

Cependant, certains protocoles comme KNX, Z-Wave, LoRaWAN/LoRa ou SigFox choisissent de ne pas utiliser ces standards. Ceci a pour effet de rendre les protocoles de l'IdO relativement hétérogènes (voir tableau 2.2).

Fort de ces caractéristiques, un fabricant doit alors choisir ses critères de sélection. Différentes stratégies s'offrent à lui mais, dans le cadre des maisons intelligentes, il ne pourra pas échapper à une poignée de protocoles : Wi-Fi/Ethernet du fait de la présence d'une box Internet et 3G/4G s'il veut que l'utilisateur de son produit puisse accéder à ses objets de l'extérieur, lorsque son smartphone n'a pas d'accès Wi-Fi. Par dessus cela, il peut en revanche utiliser d'autres protocoles, sélectionnés en fonction de leur consommation,

sécurité, accessibilité et standardisation. Du fait de la popularité de certains d'entre eux, un fabricant peut choisir de sacrifier certains critères au profit d'un protocole qui lui permettra d'avoir accès à plus d'objets connectés.

Dans le cadre des objets connectés domestiques, parmi les objets envisagés dans cette thèse, nous pouvons observer que Wi-Fi, Bluetooth et ZigBee étaient les protocoles les plus adoptés en matière de couches dédiées au transport et inférieures. KNX est également beaucoup utilisé mais demande en général à l'utilisateur d'avoir des connaissances plus poussées qu'avec d'autres protocoles qui cherchent à simplifier l'installation de produits les implémentant. Pour ce qui est des protocoles supérieurs, HTTP avec ou sans TLS est de loin le plus utilisé, suivi de DTLS et MQTT.

2.3 Vie privée

Dans cette section, j'aborderai la littérature scientifique concernant la vie privée dans les smartphones et objets connectés. Je passerai en revue les techniques utilisées pour obtenir des données, découvrir des appareils sur un réseau, détecter des fuites de données, et inférer leur contenu. Ensuite, je montrerai des fonctionnalités qui peuvent être détournées pour obtenir ou transmettre des informations. Puis je passerai aux différentes façons d'analyser automatiquement le code des applications, des parties tierces, et les interactions réseau. Enfin j'aborderai les défenses connues et la question du recueil de consentement des utilisateurs.

2.3.1 Panorama

La littérature scientifique au sujet de la vie privée dans le cadre de l'IdO est colossale. Cependant, je m'efforcerai ici de lister les principaux axes de recherche que j'ai pu rencontrer lors de ma prospection d'articles.

Certains articles ont déjà pour but de faire une prospection sur l'état actuel de la vie privée et de la sécurité dans l'IdO. Imtiaz et al. [72] abordent les menaces, attaques et techniques de défenses relatives à la vie privée dans l'IdO. Ils en tirent la conclusion que la plupart des techniques de préservation de la vie privée sont compatibles avec le RGPD mais que certaines avancées restent à faire, notamment sur les modèle d'intelligence artificielle qui devrait être considérés comme données à caractère personnel dans certains cas. Vasilomanolakis et al. [73] les propriétés particulières de ces écosystèmes (environnement non contrôlé, hétérogène, évolutif et à ressources contraintes) et Leibenger et al. [74] listent les défis qu'ils créent du fait du partage de données à caractère personnel en regard des lois européennes.

Ces articles donnent une vision globale de la situation, mais il est important de creuser les aspects techniques traités dans la littérature scientifique pour mieux comprendre quelles sont les connaissances que nous avons sur ce domaine et quels sont les principaux axes de recherche.

Je trouve également que dans ces panoramas, il n'y a pas de réflexion suffisante sur les données à caractère personnel ou sensibles. Ajouter certains modèle d'apprentissage machine est pertinent, mais quid des inférences de données en général ? Il manque aussi une réflexion sur la notion de consentement, et en particulier sur comment celui est géré lors du developpement des objets connectés et leurs applications. Le dernier article est aussi intéressant dans le sens où on constate que les utilisateurs se sentent concernés par leurs données, mais qu'ils ne sont pas prêts à payer plus pour avoir un service qui respecte leur vie privée. J'ai cependant des doutes sur la pertinence de ce genre de solution, puisque payer plus cher ne garantit pas que le service sera mieux developpé et qu'aucune erreur ne sera commise vis-à-vis des données à caractère personnel de l'utilisateur.

2.3.2 Acquisition de données, découverte d'appareils, fuites et inférences

Dans le cadre de l'IdO, une des premières étapes à franchir est de capturer des données. Cela peut être fait de plusieurs manières.

Premièrement, notamment grâce au RGPD, en téléchargeant les données qu'un acteur a stockées sur nous. Un exemple de ce genre de manoeuvre est le site Google Takeout [75] qui permet, lorsqu'on se connecte via notre compte Google, de récupérer toutes les données que Google a sauvegardé concernant notre compte.

Deuxièmement, en capturant ces données au moment ou elles sont générées ou transmises côté mobile, par exemple en instrumentalisant un smartphone sous Android avec Frida [76] afin d'analyser en temps réel les applications qui tournent dessus, voire y injecter du code. Frida permet d'ajouter des points d'arrêt dans une application sur des fonctions spécifiques et de rajouter du code à cette fonction pour modifier ou afficher ce qui se passe à l'intérieur. On peut aussi capturer des données en exploitant le système de VPN d'Android afin de capturer les données transmises par d'autres applications depuis le smartphone, par exemple en utilisant Meddle [77], Haystack [78] ou AntMonitor [79]. Cette technique consiste à créer un VPN par lequel le smartphone va passer pour communiquer, hors ce VPN étant géré par nous, on peut s'en servir pour capturer des données transmisses par les applications du smartphone avant même qu'elles quittent celui-ci.

Troisièmement, en capturant les données depuis le réseau. Certaines initiatives on été lancées pour avoir des outils de capture pratiques et portables comme PiRogue [80] qui capture le trafic réseau au moyen d'une Raspberry Pi connectée à celui-ci et tente de déchiffrer automatiquement les paquets HTTPS grâce à mitmproxy [81]. Lorsque des cas plus complexes se présentent, Burp Suite [82] est une des références en la matière. Cela permet de capturer ou intercepter du trafic, avec un nombre d'options gigantesque, comme par exemple la génération de faux certificats ou le décodage automatique de certains encodages connus. Le tout fonctionne grâce à l'ajout d'une autorité de certificat de confiance sur Android qui autorise Burp à se mettre entre l'utilisateur et le site distant. L'utilisateur peut alors envoyer des données chiffrées destinées à Burp, que ce dernier s'occupe de relayer vers le site distant, mais Burp a donc bien accès aux données en clair puisqu'il peut déchiffrer les données qui lui sont destinées. D'autres projets comme IoTInspector [83] proposent des outils d'analyse automatique du trafic, mais cette fois sans s'occuper des paquets chiffrés. C'est donc un outil dédié plus spécifiquement aux statistiques de trafic réseau.

Dans le cas particulier de la capture de trafic réseau, on cherche alors à découvrir quels sont les appareils en cours d'utilisation et à quelle catégorie d'objet ils appartiennent. Pour ce faire, IoTScanner et IoT SENTINEL [84, 85] procèdent à une phase de découverte à l'issue de laquelle on obtient une empreinte unique par appareil qui nous permet d'identifier l'appareil par la suite. Ce genre d'empreinte est obtenu grâce à des statistiques sur les paquets envoyés et reçus par des appareils (protocole, tailles, ports, ou encore ratio envoyé/reçu). Mais il est aussi possible de réaliser cette opération depuis une page HTML au moyen de scripts malicieux [86].

Une fois que les appareils sont identifiés et que les données ont été capturées, on peut procéder à l'analyse de ces dernières dans le but de détecter des données à caractère personnel, voire des données sensibles. Plusieurs articles s'attellent à l'analyse des données en clair. Wood et al. [87] cherchent des termes spécifiques venant d'un dictionnaire, et retournent les données qui les contiennent, tandis que Ren et al. [88] les analyses avec de l'apprentissage automatique.

Lorsque les données capturées sont chiffrées et indéchiffrables, il est alors nécessaire de faire de l'inférence sur celles-ci. Il est possible de se baser sur les métadonnées des

paquets, à savoir les entêtes des paquets [89] ou la tailles des paquets [90, 91, 92, 93] pour déterminer un corrélation entre une statistique particulière sur ces métadonnées et le type de paquet envoyé.

Frida reste encore, à mon avis, sous-exploité dans le sens où cet outil s'attaque aux données dès leur création dans les applications. Hors les solutions présentées ici s'intéressent aux données une fois qu'elles sont déjà créées.

2.3.3 Usage non conventionnel des systèmes

Certains préfèrent utiliser une fonctionnalité spécifique d'un objet d'une façon inhabituelle afin de faire fuiter des informations à l'insu de l'utilisateur. La diversité de ces canaux auxiliaires est relativement restreinte. Caviglione et al. [94] en fait une liste exhaustive : collusion entre applications, collusion locale (vibration, accéléromètres), utilisation de l'air (RF, acoustique, température, image, luminosité) et utilisation du réseau (altération à bas niveau).

On peut utiliser des ampoules pour transmettre des données en modulant sa luminosité [95, 96] ou, sur un principe similaire, capter, grâce au capteur de luminosité d'un smartphone, la lumière générée par un écran de smartphone pour détecter les pages que son utilisateur a consultées (les liens visités s'affichant d'une couleur différente sur un navigateur) [97]. On peut utiliser un appareil avec détecteur de mouvement pour déduire les gestes de l'utilisateur [98, 99]. Enfin, certains articles [100] utilisent les interférences d'un réseau bruité pour créer un réseau parallèle.

En plus de ces canaux auxiliaires, il est possible d'exploiter l'environnement de l'utilisateur pour déterminer son comportement, ses déplacements. Par exemple, la géolocalisation de l'utilisateur peut être détectée à son insu. Konings et al [101] la détectent en analysant les ondes d'un réseau sans-fil dans lesquelles l'utilisateur se déplace, et Arp et al. [102] en utilisant des systèmes de balises ultrasoniques captées par des applications (émission de balises captées par le smartphone en entrant dans un magasin par exemple). Ces dernières peuvent aussi être utilisées pour identifier des fraudes, comme des authentification depuis un appareil inconnu se trouvant à proximité et, par conséquent, captant les mêmes balises.

Il reste à déterminer l'impact que peuvent avoir ce genre des techniques dans la vie de tous les jours. L'exemple des balises dans les magasins est un bon point de départ, mais ce n'est certainement pas la seule occasion où un utilisateur peut se voir manipulé à son insu via des canaux auxiliaires, notamment via les smartphones qui peuvent facilement contenir des codes malicieux. Sensibiliser les gens sur ce genre de problème est important mais requiert ce genre de travaux.

2.3.4 Analyse automatique des applications et interaction réseau

L'analyse des données transmises n'est pas la seule façon de détecter des comportements suspicieux. En passant par l'analyse du code source des applications [103, 104] il est possible de détecter, par exemple, des accès à privilège excessif ou des flux de données sensibles.

En ce qui concerne l'analyse de code, on peut aussi mentionner les solutions Exodus [105] et LibRadar [106] qui permettent, à partir d'un fichier APK Android, de détecter les librairies dédiées au partage d'information avec des parties tierces.

Viennent ensuite l'analyse des appareils et celle des interactions réseau. D'autres articles s'attaquent à des analyses plus vastes, en adoptant un point de vue multidimensionnel. Ren et al. [107] s'intéressent à un grand nombre d'appareils et caractérisent les expositions d'informations selon leur destination, chiffrement, vulnérabilité aux inférences,

contenu à caractère privé ou sensible et différences régionales. De plus, Alrawi et al. [108] cherchent à dégager une méthodologie commune aux différents chercheurs, basée sur une grande base d'appareils également, afin d'analyser la sécurité de ces derniers et de partager ses résultats.

Il y a donc une opposition entre les analyses détaillées d'un faible nombre d'objets et les analyses peu détaillées d'un grand nombre d'objets. Avoir des outils communs permettant de faire des analyses poussées sur un grand nombre d'objets serait très utile, bien que beaucoup plus complexe. On pourrait imaginer un projet qui applique automatiquement une suite de techniques pour analyser les appareils, et que cette base de techniques soit augmentable librement par les utilisateurs. Les travaux d'Alrawi et al. vont dans cette directions mais je n'y trouve pas l'aspect modulable que je souhaiterais voir, pourtant essentiel. Une méthodologie est un bon point de départ, mais un outil et des algorithmes d'analyse utilisables directement serait beaucoup plus utile.

2.3.5 Défenses

Évidemment, la correction de ces problèmes est aussi traitée. Apthorpe et al. [109] montrent en 2017 qu'il est possible de bloquer le trafic réseau, dissimuler les requêtes DNS, utiliser un VPN, remodeler le trafic ou injecter du trafic. Le but de ces techniques de remodelage est de limiter la confiance d'un observateur qui chercherait à identifier des appareils ou inférer des comportements en se basant sur le trafic réseau qu'ils génèrent. Ils montrent également en 2018 que le padding de paquets est un exemple de technique de remodelage de trafic efficace [110] avec presque aucun coût sur la bande passante.

Dans une approche complètement différente, Nicolazzo et al. [111] choisissent de considérer les appareils comme des noeuds dans un réseau et de mettre en oeuvre un système de gestion, d'ajout, d'isolation de noeuds afin de protéger au maximum leurs interactions.

Les canaux auxiliaires sont complexes à corriger car ils exploitent l'environnement, donc seules des solutions physiques peuvent rendre des objets imperméables à ce genre de problème à coup sûr.

Les autres problèmes traités jusqu'ici, à savoir ceux concernant l'acquisition de données, les découvertes d'appareils, les fuites de données, les analyses automatiques et les inférences sont majoritairement dues au manque de protections du point de vue réseau, permettant à un observateur de voir ou déduire ce qu'il se passe. Certains de ces problèmes sont résolvables en renforçant la sécurité de ces réseaux, ce qui sort de la portée de cette section. D'autres sont résolvables grâce à des technique de remodelage ou de dissimulation du trafic réseau comme nous avons vu précédemment. Le tout est alors de trouver un compromis entre la quantité de mécanismes à appliquer, et l'impact sur la bande passante que ces mécanismes auront.

Il reste tout de même un point important rarement traité : la prise en compte du format des données dès la conception des protocoles utilisés. Par exemple un format binaire sera plus compliqué à interpréter par un observateur qu'un format JSON. Envoyer des trames de taille aléatoire ou de taille constante permet aussi de réduire énormément le nombre d'inférences possibles sur ces paquets. Un problème très complexe du point de vue du compromis entre efficacité et bande passante est celui de la détection d'activité. En effet, peu importe les mécanismes de défense contre les inférences qu'on puisse appliquer, une activité générera toujours des paquets sur le réseau. Il est possible de résoudre ce problème en générant du trafic à des moments aléatoires, mais cela crée du trafic inutile, du bruit qui peut impacter la bande passante. Il faut donc choisir entre vie privée renforcée ou bande passante optimale.

2.3.6 Consentement et comportements utilisateurs

Parmi les vecteurs de fuite de données à caractère personnel, Li et al. [112] listent le mauvais usage des objets et applications de la part de l'utilisateur, leur tendance à choisir des applications connues qui ont un intérêt particulier à la capture de leur données.

Plusieurs études montrent que les connaissances des utilisateurs et leur aptitude à se défendre contre les manipulations des fabricants sont faibles [113, 114, 114].

Une première raison est la tendance des utilisateurs à ne pas lire les chartes de vie privée. Pour pallier à ce problème, une solution présentée par Shayegh et al. [115] propose de collecter des chartes de vie privée et d'en extraire les passages concernant des données sensibles afin de réduire la taille des textes à lire, et de classifier ces extraits en fonction de leur sujet, ceci afin d'inciter les utilisateurs à lire et prendre connaissance des chartes de vie privée de ses objets.

Liu et al. [116] ont montré que la suggestion de configurations respectueuses de la vie privée pouvait mener à de meilleurs usages des applications, puis Stach et al. [117] on proposé comme solution de recommander automatiquement des configurations sur mesure grâce à l'apprentissage automatique.

Cependant les développeurs, au moment de la conception des objets et applications sont aussi atteints de biais. Perez et al. [118] ont par conséquent montré qu'il est aussi possible de les limiter au moyen de lignes directrices dédiées.

En l'absence de protection, les utilisateurs pourraient être bien ennuyés dans certaines situations. Par exemple, les assurances et tribunaux pourraient avoir un intérêt particulier à récupérer leurs données stockées dans les objets connectés en guise de preuves d'un comportement particulier soit disant mauvais [119].

2.4 Sécurité

Dans cette section, j'aborderai la littérature scientifique concernant la sécurité dans les smartphones, et celle dans les objets connectés. Je commencerai par les vulnérabilités connues dans les objets connectés et leurs applications, puis dans les protocoles et technologies de connectivité qu'ils utilisent. Je poursuivrai par un type d'attaque particulièrement étudié dans le cadre des assistants personnels et des enceintes intelligentes : les injections de commandes. Suite à cela, j'aborderai l'aspect de la sécurité sanitaire et, enfin, les défenses connues aujourd'hui.

2.4.1 Panorama

À l'instar de la vie privée, la sécurité dans l'IdO a son lot de littérature, qui peut être défrichée au moyen de quelques articles résumant son état actuel. Pongle et al. [120] abordent des attaques réseau sur 6LoWPAN, en les catégorisant dans plusieurs catégories (attaques par fragmentation, authentification, confidentialité, et menace depuis l'Internet).

Krejci et al. [121] abordent des attaques réseau sur BLE, LoRaWAN, ZigBee et Z-Wave.

Ces articles couvrent une grande partie des problèmes de sécurité rencontrés dans les protocoles de l'IdO. L'aspect sécurité des protocoles n'étant pas le sujet de cette thèse, je me contenterai ici de commenter ce que contiennent ces articles, mais beaucoup d'autres sont disponibles, dont certains que je présenterai dans les sections suivantes.

Les deux articles donnent beaucoup d'informations sur la sécurité de certains protocoles qui peuvent être utiles pour l'analyse de ceux-ci sous l'angle de la vie privée. Par exemple des attaques sur l'authentification ou la confidentialité de 6LoWPAN peuvent permettre d'obtenir des informations sur les informations transitant dans le réseau, pourvu

qu'IPsec ne soit pas utilisé. Le système d'échange de clés de Bluetooth par défaut est également exploitable et ne fourni pas de sécurité suffisante contre les attaques de l'homme du milieu ou de rejeu. Il est aussi possible de récupérer le contenu en clair des trames LoRaWAN avec une combinaison d'attaques. Enfin ZigBee est connu pour avoir utilisé des clés maître préinstallées qui ont été découvertes en 2015.

2.4.2 Vulnérabilités des objets, applications, protocoles et technologies de connectivité

L'IdO est réputé être un domaine récent et hautement susceptible d'être conçu sans préoccupation de la sécurité a priori. Chothia et al. [122] utilisent même ce désavantage comme opportunité dans certains contextes comme l'apprentissage de la sécurité informatique dans l'enseignement.

D'autres peuvent profiter de cette opportunité, comme les rançongiciels qui vont très probablement tirer avantages de ces objets pour se disperser et créer de nouvelles menaces [123]. Les cryptomonnaies auront aussi tout avantage à prendre partie des objets en tant que puissance de calcul [124]. La possibilité de faire des attaques de déni de service généralisées [18] est également un des effets redoutés de la vulnérabilité des objets connectés.

Ces problèmes de conception se retrouvent dans tout le spectre des objets connectés, mais certains appareils reçoivent une attention plus forte du fait de leur notoriété, comme le Fitbit par exemple [125], du fait de leur ubiquité [126] ou du fait de leur sensibilité [127].

Ikram et al. [128] montrent aussi que certaines applications sont également capables d'exploiter des privilèges Android pour rediriger du trafic sur tunnel VPN et espionner des transmissions d'autres applications.

Viennent ensuite la sécurité des protocoles et technologies de connectivité. Il y a une myriade d'articles qui traitent de problèmes liés à un protocole en particulier. Entre autres, Vanhoef et al. [129] a dévoilé des vulnérabilité du Wi-Fi en parvenant à compromettre le chiffrement WPA2, Ryan [130] a créer une suite d'outils logiciels et matériels afin d'exploiter le système de channel hopping du Bluetooth. Cao et al. [131] s'attaquent au ZigBee qui peut permettre de vider l'énergie des objets l'implémentant, et Ronen et al. [132] parviennent à générer des réactions d'extinction en chaîne d'ampoules utilisant ce protocole. Z-Wave, n'est pas en reste puisque son niveau de sécurité lors de l'appairage peut être réduit [133]. Enfin Avoine et al. [134] décrivent des attaques de déconnexion de force des objets, ou de rejeu et de déchiffrement de paquets sur le protocole LoRaWan.

Il est clair qu'aucun protocole n'est irréprochable, mais il ne faut pas perdre de vue que leurs objectifs sont différents, ce qui fait que, par exemple, appliquer un chiffrement de paquets peut être un choix difficile à prendre pour un protocole destiné à utiliser le moins d'énergie possible. Il faut aussi noter que ces protocoles reçoivent parfois des correctifs et des améliorations, comme l'implémentation de WPA3 pour Wi-Fi par exemple. Il faut donc faire son choix parmi ces protocoles selon ce qu'on attend d'eux. Certains seront aussi quasiment obligatoires si l'usage d'Internet est nécessaire au bon fonctionnement d'un objet connecté, comme Wi-Fi par exemple.

Si la sécurité et la vie privée sont les objectifs à privilégier, Wi-Fi, ZigBee et Thread semblent être de bons compromis puisqu'ils ne sont pas contraints par la bande passante et par les ressources matérielles comme d'autres protocoles tels que LoRaWAN, ce qui laisse plus de libertés pour implémenter des mécanismes de sécurité robustes. Bluetooth peut aussi être personnalisé suffisamment pour qu'on puisse ajouter des mécanismes de sécurité supplémentaires, bien que les implémenter soit même soit un vecteur potentiel de nouvelles vulnérabilités.

En ce qui concerne les objets connectés et leurs applications, on n'a pas d'autre choix

que de faire confiance aux dires des fabricants. Un fabricant qui semble plus sensible à la question de la vie privée prouve au moins qu'il connait la problématique et qu'il est capable d'agir dans le bon sens. La meilleure solution reste encore de faire un audit de leurs produits pour se faire un avis objectif.

2.4.3 Injection de commandes

Une des vulnérabilités les plus étudiées, que ce soit sur les enceintes intelligentes ou les assistants vocaux, est l'injection de commande [135, 136, 137, 138, 139]. Elles fonctionnent sur le principe d'ajout de bruit au signal audio, de telle sorte que la commande que l'utilisateur entend soit inaudible, invisible, ou audible mais embarquant une commande cachée.

Diao et al. [140] s'attaquent en particulier aux émissions de son via une application pour déclencher une commande de l'assistant vocal exécutée sur le smartphone même.

Enfin, Villalba et al. [141] apportent quelques solutions, tentant de détecter les flux audio modifiés ou compromis, afin d'empêcher leur exécution.

2.4.4 Défenses

Du point de vue des techniques de défense, chaque attaque, peu importe sa cible, reçoit avec le temps son lot de correctifs apportés par les fabriquants des objets eux-même. Il est aussi maintenant courant de voir des « bug bounties » pour permettre aux hackers du monde entier de trouver des vulnérabilités dans des systèmes contre une somme d'argent proportionelle à la gravité de la vulnérabilité découverte. En revanche, ces plateformes sont surtout axées sur les problèmes de fonctionnalités ou de sécurité. Comme nous avons pu le voir avec Google lors de nos travaux, les problèmes liés à la vie privée sont encore assez peu considérés.

À côté de cela, certains proposent de considérer les problèmes de sécurité liés au développement dès la conception. Apvrille et al. [142] mettent en avant des lignes directrices de développement pour concevoir des objets sécurisés dès la conception. Ces lignes directrices permettent d'évaluer les risques, les objectifs de sécurité associés, et d'implémenter les mécanismes de sécurité appropriés lors de la conception, du développement et de la phase de tests. Elles suggèrent également plusieurs sources courantes de problèmes à observer tout particulièrement.

D'un point de vue légal, certaines mesures sont appliquées pour imposer une meilleure sécurité aux fabricants. Par exemple, en Californie une loi a été adoptée [143] pour imposer aux fabricants d'attribuer un mot de passe par défaut unique pour chacun des objets connectés qu'ils vendent. Le RGPD encourage également à « garantir un niveau de sécurité adapté au risque numérique » et l'ANSSI met à disposition « de nombreux supports et outils afin d'œuvrer au renforcement de la sécurité des données à caractère personnel et de leur sécurité numérique au sens large » [144].

Même si ces aides sont fournies, il reste plus difficile pour les petits fabricants de sécuriser leurs produits que pour les grands fabricants qui sont eux-même faillibles. Il parait alors important de trouver un moyen de permettre aux plus désavantagés d'atteindre un niveau de sécurité équivalent à celui des plus avantagés.

2.5 Autres considérations non traitées

L'Internet des objets est un sujet très vaste et je ne traiterai pas de toutes ses problématiques dans cette thèse. Cependant quelques points ont attiré mon attention, et qui méritent d'être mentionnés ici.

Les applications et leurs vulnérabilités sont un axe de recherche à part entière et très vaste qui n'est pas traité dans cette thèse. Il existe néanmoins quelques articles qui m'ont inspiré pour les travaux de cette thèse. Notamment, Wang et al. [145] ont montré que les applications officielles liées aux objets peuvent être un grand vecteur de vulnérabilités du fait des composants logiciels qu'elles peuvent partager, provoquant des vulnérabilités en masse et identifiables automatiquement.

Un point tout aussi important, est celui de l'impact de l'Internet des objets sur les personnes. Ronen et al. [96] abordent les problématique de la santé en suggérant qu'il est possible de déclencher des crises d'épilepsie grâce à des ampoules connectées.

Des questions éthiques se posent également sur des aspects inattendus comme le don d'organe qui pourrait être moins efficace si les voitures autonomes contribuaient à significativement réduire l'accidentologie, grande pourvoyeuse d'organes sains [146].

L'éthique et les impacts potentiels de l'IdO sur le monde sont des sujets primordiaux mais complexes. J'aborderai également succintement dans cette thèse l'aspect énergétique de l'IdO qui mérite d'être urgement étudié selon moi au vu des conditions écologiques actuelles.

2.6 Positionnement de cette thèse

La littérature scientifique se concentre avant tout sur des objets en particulier, et négligent leurs techniques de contrôle. Pourtant, comme je le montrerai dans les chapitres suivants, l'étude des différentes façons de contrôler un même objet est cruciale lorsqu'il s'agit de protéger la vie privée de l'utilisateur et la sécurité des appareils, applications et protocoles. Même en étant sécurisé, un objet est tributaire de ses différentes techniques de contrôle.

Nous savons déjà que certaines attaques sont possible, comme l'inférence des données ou la fuite d'information dans les parties tierces. Nous savons aussi que certaines données à caractère personnel peuvent être injustement capturées par certains acteurs et que ces données peuvent parcourir des chemins très complexes. Mais il est important de pousser ces connaissances plus loin en adoptant, encore une fois, le point de vue des techniques de contrôle, qui soulèvent les mêmes problématiques, avec une complexité d'autant plus forte tant il y a de possibilités d'interactions, de configurations et d'acteurs.

Certaines questions sont aussi peu posées, comme celle de la souveraineté et celle de la capacité des utilisateurs à donner un consentement libre, spécifique, éclairé et univoque. Également, et en sortant du champ de ces travaux, la question de l'économie d'énergie devrait être posée.

C'est donc sur ces problématiques que j'ai choisi d'échafauder cette thèse. Dans le chapitre 3 je présenterai les problèmes liés aux techniques de contrôle : étude des chemins de données, transmissions périodiques, problèmes de souveraineté, inférences d'actions, fuites de données ou encore traqueurs d'applications.

Dans le chapitre 4, je présenterai les problèmes liés aux configurations des applications vis-à-vis du consentement ainsi que les problèmes de consentement liés aux chartes de vie privée.

Je terminerai par une conclusion et des perspectives dans le chapitre 5.

Chapitre 3

Allumer et éteindre une ampoule : des conséquences pas si anodines

Ce chapitre est tiré de l'article en cours de soumission dont la pré-publication se trouve sur HAL [147] et dont je suis l'auteur principal.

3.1 Introduction

Dans ce chapitre, nous nous concentrons sur la classe d'objets connectés des ampoules intelligentes et considérons trois produits représentatifs et populaires : d'une part l'ampoule Philips Hue White [148] et l'ampoule IKEA Tradfri [149], toutes deux connectées en Zigbee à un dispositif de pontage Philips ou IKEA, d'autre part une ampoule LIFX Mini White [150] qui se connecte directement au réseau domestique Wi-Fi. Nous considérons un utilisateur final vivant en France (le RGPD s'applique aux résidents de l'Union européenne, quelle que soit leur nationalité). Ce travail vise à évaluer la partie commande de ces appareils, utilisée pour l'allumer ou l'éteindre.

À la maison, nous supposons que l'utilisateur utilise soit un smartphone (ou une tablette) connecté au réseau Wi-Fi domestique, soit un haut-parleur intelligent, soit un bouton intelligent. Plusieurs applications sont disponibles sur le smartphone de l'utilisateur qui peuvent être utilisées à cette fin, provenant de plusieurs développeurs ou sociétés. Lorsqu'il est distant, l'utilisateur utilise son smartphone avec soit une connexion cellulaire 4G, soit une connexion à un réseau Wi-Fi connu, et l'une des applications qui peuvent fonctionner à distance. L'évaluation est centrée sur les considérations relatives à la vie privée, en se concentrant en particulier sur les chemins de données : qui est informé de quoi chez l'utilisateur ?

Nos principales contributions sont les suivantes :

— Ce travail adopte un point de vue original dans lequel nous nous concentrons essentiellement sur la partie contrôle des ampoules : Il montre que des différences majeures sont dues à la technique utilisée pour contrôler les ampoules, et donc que de nombreuses observations et conclusions que nous tirons restent valables pour un large ensemble d'appareils.

— Nous soulignons les principales préoccupations en matière de vie privée et de souveraineté : Par exemple, la plupart des scénarios (28 sur 33) impliquent la transmission des données de l'utilisateur à Google ou à Amazon, même si l'utilisateur est chez lui. La présence de transmissions périodiques à des serveurs distants permet également un accès permanent à des appareils internes, parfois par des serveurs situés dans un pays non européen.

— Nous montrons que déduire l'action de l'utilisateur en analysant la taille d'une requête est plutôt efficace, même avec un trafic chiffré.
— Nous montrons que la situation est trop complexe pour que l'utilisateur final puisse comprendre les implications en matière de vie privée des différents choix qui s'offrent à lui pour contrôler son ampoule intelligente.

Il convient de noter que ce travail ne prend pas en compte la configuration de l'équipement avant toute action de commutation des ampoules, à savoir l'initialisation du pont Philips ou de la passerelle IKEA, des ampoules et des applications que nous avons examinées. Cet aspect est laissé pour les travaux futurs.

Le chapitre est organisé comme suit. Nous décrivons d'abord la méthodologie expérimentale et les outils conçus dans la section 3.2. Nous poursuivons avec une analyse des résultats expérimentaux dans la section 3.3, en nous concentrant sur les chemins de données, les implications en termes de souveraineté, les inférences possibles par l'analyse de la taille des messages et les informations divulguées dans les messages ou par les traqueurs d'applications. Nous terminons par un rappel des travaux connexes dans la section 3.4 et de nos contributions par rapport à ceux-ci, une conclusion dans Section 3.5, et nous résumons les résultats de notre divulgation responsable.

3.2 Méthodologie

3.2.1 Buts du travail

Notre travail vise à évaluer l'impact sur la vie privée de la partie contrôle de trois ampoules intelligentes représentatives et populaires présentes dans de nombreuses maisons intelligentes. Le choix des ampoules intelligentes est intéressant car les utilisateurs finaux peuvent sous-estimer leurs implications sur la vie privée (par exemple, par rapport à une caméra à reconnaissance faciale). En fait, connaître l'état de toutes les ampoules d'une maison peut facilement révéler les habitudes des habitants, leurs périodes d'activité quotidienne et les schémas réguliers ou exceptionnels. En soi, ces informations révèlent des renseignements personnels et, croisées avec d'autres sources d'information, elles contribueront à établir un meilleur profil utilisateur. Enfin, l'appareil lui-même ne produit pas un volume important de données, ce qui signifie que les volumes de données que nous observons sont principalement dus à la partie contrôle elle-même. Nous nous concentrons sur un utilisateur final français par commodité, mais les conclusions s'appliquent également aux résidents de l'Union Européenne pour lesquels le RGPD s'applique.

Les expériences portent sur deux situations différentes. Tout d'abord, nous considérons un utilisateur à domicile, qui utilise soit un smartphone (ou une tablette) connecté au réseau Wi-Fi local, soit une enceinte intelligente, ou encore un bouton intelligent. Ensuite, nous considérons un utilisateur à distance, physiquement ou logiquement (par exemple, en utilisant son smartphone et une connexion 4G), et évaluons si un contrôle à distance est possible ou non, et les conséquences liées. L'une des questions qui nous intéressent est de savoir si une entreprise a trop privilégié la possibilité de contrôler à distance une ampoule, une fonction non essentielle, en oubliant d'avoir un contrôle purement local lorsque cela suffit, car la minimisation des données est une caractéristique essentielle de la vie privée et, en cas de coupure de la connexion Internet, un besoin pratique essentiel. Il est évident que le recours à des services à distance peut être justifié par des avantages objectifs clés, par exemple, l'utilisation de technologies avancées de TALN (Traitement automatique du language naturel) et le croisement de données entre les différents services, mais ce choix a de nombreuses conséquences, notamment en termes de vie privée.

3.2.2 Environment de test

Table 3.1: Liste des appareils, version de firmware et connectivité.

Appareil	Version firmware	Connect.
Amazon Echo Spot [151]	647591020	Wi-Fi
Google Home [152]	171861	Wi-Fi
Passerelle IKEA Tradfri [153]	1.9.27	Eth. + ZigBee
Télécommande IKEA Tradfri [154]	2.3.014	ZigBee
Ampoule IKEA Tradfri [149]	1.2.217	ZigBee
Smartphone LG Nexus 5 [155]	Android 6.0.1	Wi-Fi
Ampoule LIFX Mini White [150]	3.50	Wi-Fi
Pont Philips [156]	1935074050	Eth. + ZigBee
Ampoule Philips Hue white [148]	1.46.0_r26312	ZigBee
Raspberry Pi 3 Model B [157]	Home Assistant 2.12 /openHABian 2.4.0-1	Eth.

Table 3.2: Liste des applications smartphone Android et leur version.

Application	Version
Amazon Alexa [158]	2.2.293218.0
All4Hue [159]	9.2
Google Home [160]	2.13.50.15
Chrome (pour Home Assistant) [161]	76.0.3809.132
Philips Hue [162]	3.27.0
Hue Hello [163]	1.34
IFTTT [164]	4.1.0
IKEA Home smart [165]	1.10.1
LIFX [166]	3.15.2
openHAB [167]	2.8.0

Le tableau 3.1 liste les différents composants physiques de la plateforme et le tableau 3.2 liste les applications pour smartphones Android : les trois ampoules sont fournies avec leur application officielle, par le fabricant de l'ampoule ; Amazon Alexa est l'application officielle pour gérer l'Echo Spot et possède sa propre interface pour allumer et éteindre les différentes ampoules ; Il en va de même pour l'application Google Home ; All4Hue et Hue Hello sont deux applications non officielles pour l'écosystème Philips ; IFTTT est une application destinée à l'IdO en général qui intègre une variété de widgets, y compris pour les ampoules que nous considérons (nous avons utilisé les recettes `Philips Hue Turn on/off your lights with one tap on your phone` et `LIFX Toggle LIFX lights on/off`) ; openHAB est une application open-source qui se connecte directement à un serveur openHAB [168] hébergé sur une Raspberry Pi ; Home Assistant [169] est une autre solution open-source, également hébergée sur une Raspberry Pi, et comme elle n'a pas d'application, nous avons utilisé Chrome pour accéder au serveur web de Home Assistant hébergé sur le Raspberry Pi.

La configuration expérimentale est illustrée dans la figure 3.1. Elle comprend un ordinateur portable (Dell Latitude E6410 fonctionnant sous Linux) faisant office de box d'un

fournisseur d'accès à Internet (FAI) : il relaie le trafic vers/depuis Internet, il fournit l'interface Wi-Fi `wlan0` (il utilise un adaptateur Wi-Fi TPE-N150USB USB sans fil Penguin en mode point d'accès), et l'interface Ethernet câblée `eth0`. Les interfaces Wi-Fi `wlan0` et Ethernet `eth0` sont toutes deux utilisées pour assurer la connexion sur le réseau local domestique. L'interface `eth0` est connectée à un concentrateur Ethernet (plutôt qu'à un commutateur) afin de permettre le reniflage du trafic de l'interface Ethernet sur les différents ports de la box simulée, et un point d'accès Wi-Fi avec son propre SSID est défini sur l'interface `wlan0`.

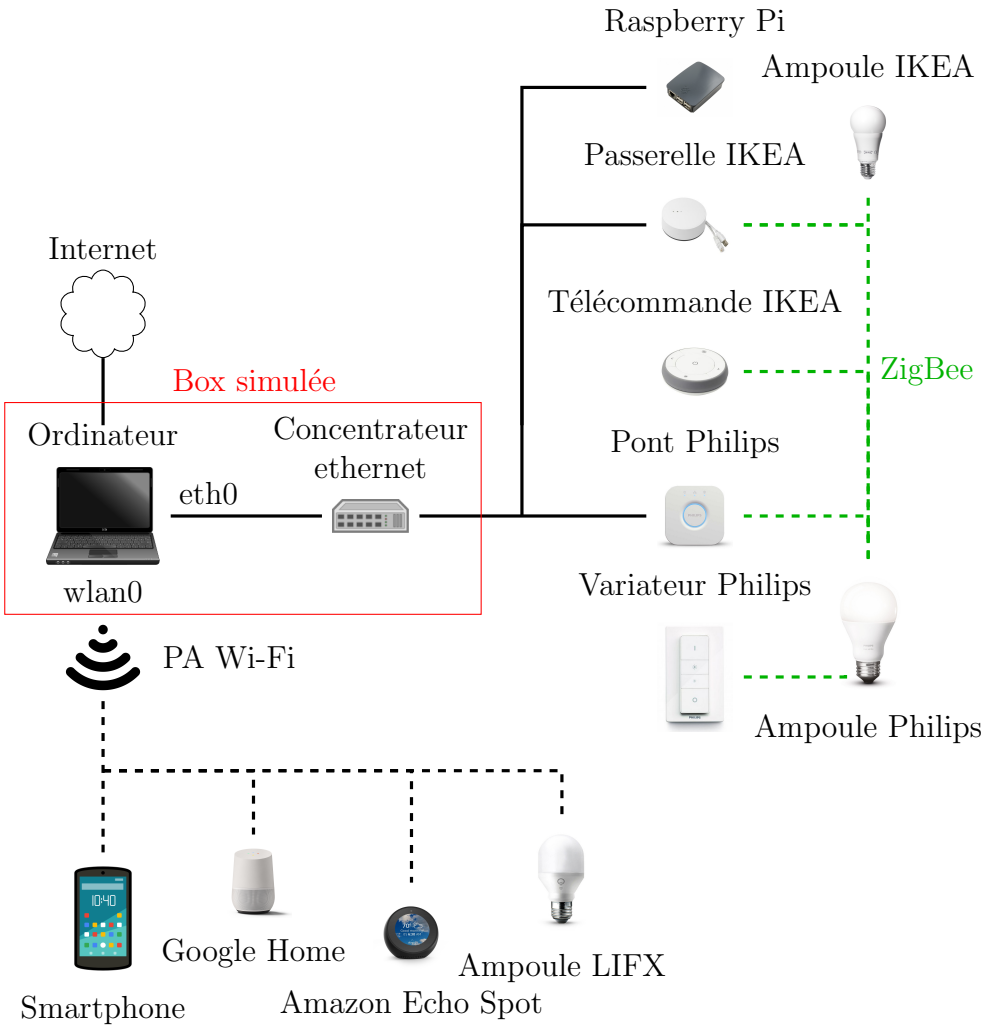


Figure 3.1: Configuration de base. *L'ordinateur sert à relayer les communications entre Internet et les appareils connectés au concentrateur Ethernet ou au point d'accès Wi-Fi. La combinaison de l'ordinateur, du concentrateur et du point d'accès simule une box de FAI.*

Pour traiter la question du réacheminement d'Internet et pour que les deux réseaux `eth0` et `wlan0` soient sur le même sous-réseau, nous avons utilisé un ensemble de règles `iptable` et de règles de pont logiciel, ainsi que des configurations spécifiques `wpa_supplicant`, `hostapd` et `dhcpd`. Cette configuration (voir Section 3.2.7) permet de connecter des appareils sur le concentrateur ou sur le point d'accès, et leur permettre de communiquer entre eux et avec Internet de manière transparente.

Cette configuration permet de capturer tout le trafic (local, vers et depuis Internet) en utilisant Wireshark. La seule exception est le réseau ZigBee entre le dispositifs de

pontage Philips/IKEA et les ampoules Philips/IKEA qui n'est pas surveillé. Ce n'est pas un problème car nous sommes principalement intéressés par le trafic Internet plutôt que par le trafic local ZigBee entre les appareils.

3.2.3 Captures et méthodologie

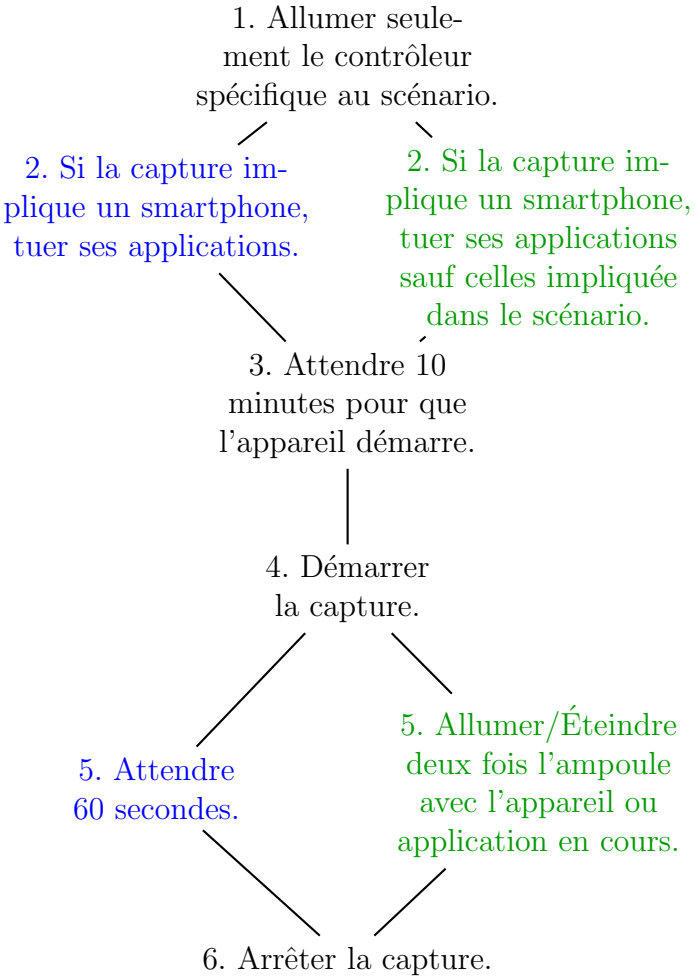


Figure 3.2: Méthodologie des captures. *Sont décrites : à gauche (bleu) la méthodologie utilisée pour les captures de référence, à droite (vert) celle des captures d'usage et au milieu (noir) les étapes communes.*

Toutes les captures se répartissent en deux catégories : « référence » ou « usage ». Ces catégories distinguent le trafic réseau par défaut (captures de référence) du trafic généré lors de l'allumage ou de l'extinction des ampoules (captures d'usage). Toutes les captures ont été effectuées à partir de l'interface `Any` interface[1] disponible sur Wireshark afin de voir toutes les communications en cours, que ce soit d'appareil à appareil ou d'appareil à Internet et vice-versa.

La figure 3.2 détaille la méthodologie suivie lors des captures de référence (à gauche) et d'usage (à droite). Le temps d'attente de 10 minutes (étape 3) est nécessaire pour garantir que tous les appareils atteignent un état stable après avoir été mis en marche (les expériences antérieures ont montré que cela peut prendre plusieurs minutes). Pour

1. Wireshark permet de réaliser des captures sur toutes les interfaces et fournit une interface appelée `Any (Linux cooked)` qui les fusionne en une seule.

les captures de référence, le temps d'attente de 60 secondes (étape 5) a été choisi de manière à conserver un temps de capture total raisonnable tout en permettant de voir les transmissions périodiques de certains appareils. Avec les captures d'usage, nous allumons et éteignons l'ampoule deux fois (c.-à-d. ON – OFF – ON – OFF) pour augmenter la précision des mesures et réduire le risque de rater des transmissions périodiques.

3.2.4 Les 8 scénarios de références et 33 scénarios d'usage

La méthodologie de la figure 3.2 a été appliquée à divers scénarios, chacun d'eux étant lié à une manière spécifique de contrôler l'ampoule (capture d'usage), ou servant de référence (capture de référence). Nous avons donc effectué une capture par scénario [2]. Un autre objectif que nous avions initialement envisagé pour les captures de référence était de constater les différences de comportement lors de l'intégration de nouveaux appareils dans le réseau ou d'une quelconque concurrence entre les appareils. Cependant, nous n'avons pas été témoins de telles situations et nous n'avons conservé que 8 scénarios de référence utiles impliquant respectivement le pont Philips, la passerelle IKEA, l'ampoule LIFX, le variateur Philips, la télécommande IKEA Tradfri, l'Amazon Echo Spot, le Google Home et le smartphone.

Table 3.3: Les 33 scénarios d'usage. *Dans les cas de Home Assistant et openHAB, le scénario implique à la fois le smartphone et une Raspberry Pi qui exécute le logiciel associé.*

	Technique de contrôle	Peut utiliser : Pont Philips	Passerelle IKEA	Ampoule LIFX
Appareils	Amazon Echo Spot	✓	✓	✓
	Google Home	✓	✓	✓
	Variateur Philips	✓	n/a	n/a
	Télécommande IKEA Tradfri	n/a	✓	n/a
Applications smartphone	Alexa	✓	✓	✓
	Alexa avec mic.	✓	✓	✓
	All4Hue	✓	n/a	n/a
	Home	✓	✓	✓
	Home avec mic.	✓	✓	✓
	Home Assistant + R. Pi	✓	✓	✓
	Philips Hue	✓	n/a	n/a
	Hue Hello	✓	n/a	n/a
	IFTTT	✓	n/a	✓
	IKEA Home smart	n/a	✓	n/a
	LIFX	n/a	n/a	✓
	openHAB + R. Pi	✓	✓	✓

En ce qui concerne les scénarios d'usage, le tableau 3.3 liste les appareils et les applications pour smartphones que nous avons examinés ainsi que la manière dont ils sont utilisés : en utilisant la voix ou l'interaction physique avec un appareil (c.-à-d. les enceintes intelligentes et les boutons intelligents), en utilisant le microphone du smartphone par l'in-

2. Nous avons parfois répété ces captures, par exemple dans la section 3.3.3 pour accroître la confiance en les tailles de trame ON et OFF.

termédiaire de l'application, ou en utilisant un bouton ON/OFF dans l'application. Au total, nous avons examiné 33 scénarios.

Il est important de noter que les solutions Home-Assistant et openHAB impliquent toutes deux la Raspberry Pi (interface web accessible par le navigateur du smartphone avec Home-Assistant, ou via une application dédiée au smartphone avec openHAB). Les captures effectuées pour ces scénarios recueillent donc des données provenant du pont, du smartphone et de la Raspberry Pi. Sinon, la Raspberry Pi n'est pas utilisée.

3.2.5 Nos outils de statistiques et labellisation

Afin de faciliter l'interprétation des captures, nous avons développé des outils pour calculer des statistiques sur ces dernières. Notre outil Python `pcapstat`, que nous avons rendu public (voir section 3.2.7), effectue une capture et résume plusieurs paramètres, tels que le nombre de trames, le fabricant des appareils ou la source et la destination. Il essaie de déduire automatiquement certaines informations comme l'identité d'une source ou d'une destination en utilisant les programmes `whois`, `geoipify`[3] et `host`, et en exploitant les requêtes DNS qui sont détectées. Chaque source et destination se voit attribué un label lisible et compréhensible et toutes les données générées par `pcapstat` sont exportées et prêtes pour un post-traitement en utilisant le framework Pandas [170].

Le label que nous utilisons pour les sources et les destinations est ce que nous appelons le « meilleur label connu ». Il consiste en une adresse IP dans le pire des cas. Sinon, plus nous déduisons de renseignements, plus nous améliorons ce label. Plusieurs informations sont potentiellement utilisées, à savoir le `whois` de l'adresse IP, le code pays du serveur, le nom d'hôte du serveur (son « FQDN ») renvoyé par la commande `host`, ou le nom exact si une requête DNS a été trouvée dans la capture vers cette IP particulière. Voici quelques exemples :

- `(US) AT-88-Z/ ip : 52.95.121.5 (IP/TCP)`
- `(US) GOOGLE/ rdns : time1.google.com (IP/UDP/NTP)`
- `(US) GOOGLE/ drdns : www.google.com (IP/TCP)`

Le premier exemple concerne le cas d'une IP, d'un pays, d'une organisation (« Amazon Technologies ») et d'un protocole (seuls `whois` et `geoiplookup` ont abouti). Le deuxième ajoute un mot-clé `rdns` (Reverse DNS), ce qui signifie que la commande `host` a réussi à trouver un nom d'hôte. La troisième donne un `drdns` (Detected Reverse DNS), ce qui signifie que nous avons trouvé une requête DNS à l'intérieur de la capture vers ce nom d'hôte, ce qui améliore la fiabilité (en effet, plusieurs noms d'hôtes pour différents services peuvent être associés à la même adresse IP, ce qui crée une ambiguïté). Le fait de disposer de labels expressifs et d'extractions automatisées selon plusieurs points de vue s'est avéré essentiel dans l'analyse de nos captures nombreuses et complexes.

3.2.6 Accéder aux données en clair des paquets

Nous nous sommes également intéressés au contenu des paquets. Lorsque les paquets sont en texte clair, nous prélevons leur contenu, en supprimant tout caractère ASCII non lisible. Le cas de LIFX est un peu particulier car LIFX utilise l'UDP pour transmettre des données en utilisant son propre protocole. La documentation du protocole nous a permis d'extraire le contenu exact des paquets LIFX (par exemple, pour savoir s'il s'agit d'une requête ON ou OFF).

3. Bien que cet outil de géolocalisation nous donne une bonne idée de l'endroit où se trouve l'adresse IP, il ne peut pas être considéré comme totalement fiable : la localisation pourrait être obsolète ou l'adresse IP pourrait nous conduire à différents serveurs dans différents lieux en fonction, par exemple, de l'endroit où nous nous trouvons lors de la requête.

Heureusement d'un point de vue sécurité et vie privée, de nombreux paquets sont chiffrés. Dans ce cas, nous lançons une attaque de type « man-in-the-middle », en utilisant plusieurs outils. Avec les applications pour smartphones, le premier outil est `Burp Suite` [82] qui intercepte les trames envoyées par une application et les déchiffre lorsque c'est possible. L'utilisateur du smartphone doit définir `Burp` comme proxy dans les « paramètres Internet » du smartphone et ajouter l'autorité de certification de `Burp` aux certificats de confiance Android. Après cela, `Burp` est approuvé par Android, il génère son propre certificat à la place du serveur distant et prend le rôle de l'application pour le serveur. Ainsi, chaque trame passe par `Burp` et est déchiffrée de manière transparente. `Burp` permet également de rejeter ou de transférer tout paquet qu'il reçoit, ce qui est pratique pour identifier quel paquet déclenche l'action ON ou OFF sur une ampoule.

Cependant, certaines applications utilisent l'épinglage de certificats (certificate pinning) et ne font pas confiance à un certificat qui n'est pas défini dans leur code. Pour contourner ce problème, nous avons utilisé un deuxième outil, `Frida`, ainsi qu'un script [171] destiné à désactiver cet épinglage. Grâce à ce script, nous avons pu accéder au texte en clair des paquets d'Alexa, la seule application parmi celles considérées qui utilisait l'épinglage des certificats.

Ces techniques nous ont permis d'obtenir le texte en clair des paquets pour la plupart des applications, à l'exception des applications Philips Hue, Hue Hello et IKEA Tradfri. Les raisons de ces échecs ne sont pas claires, mais elles pourraient être dues à la détection de proxy ou à des mécanismes de contournement, ou à l'utilisation de DTLS au lieu de HTTPS pour Tradfri.

En ce qui concerne les objets connectés, à l'exception de l'ampoule LIFX, nous n'avons pas réussi à déchiffrer les paquets. En effet, `Burp` exige de l'utilisateur qu'il ajoute une nouvelle autorité de certification, celle de `Burp`, dans l'appareil, ce qui n'était pas possible ici.

3.2.7 Préoccupations éthiques et reproductibilité

Toutes les captures ont été effectuées dans un environnement contrôlé, à l'aide de noms d'utilisateur et de courriels inventés, sans qu'aucun utilisateur réel ne soit impliqué.

Tous les outils, les captures brutes et les détails de configuration seront rendus publics avant la publication.

3.3 Résultats expérimentaux

Cette section détaille les résultats expérimentaux. Nous commençons par une analyse macroscopique, en considérant les chemins de données, le volume de données échangées sur Internet et les considérations de souveraineté. Dans un deuxième temps, nous examinons les messages eux-mêmes, comment leur taille permet souvent de déduire les actions de l'utilisateur malgré le chiffrement, le contenu des messages, ainsi que la présence de traqueurs dans les applications des smartphones. Nous terminons par une analyse des caractéristiques bénéfiques pour la vie privée et évaluons si elles sont ou non utilisées dans les différentes techniques de contrôle.

3.3.1 Taxonomie des chemins de données

Le cas des scénarios de référence

Concentrons-nous d'abord sur les scénarios de référence. Le tableau 3.4 montre que dans tous les scénarios impliquant le pont Philips, la passerelle IKEA avec intégration et

Table 3.4: Transmissions périodiques de paquets pour chaque appareil.

Pont Philips		
Paquets NTP	à time.google.com[1]	chaque 30s
Paquets TCP Keep-Alive	à ws.meethue.com	chaque 30s
Paquets TLS (95 octets)	à ws.meethue.com	chaque 120s
La passerelle IKEA sans intégration n'envoie rien		
Passerelle IKEA avec intégration Alexa/Google Assistant		
Paquets TCP Keep-Alive	à amazonaws.com[2]	chaque 30s
Paquets TLS (87 octets)	à amazonaws.com[2]	chaque 30s
Ampoule LIFX		
Paquets TCP Keep-Alive	à v2.broker.lifx.co	chaque 45s
Paquets TLS (173 octets)	à v2.broker.lifx.co	chaque 90s
Paquets TLS (121 octets)	à v2.broker.lifx.co	chaque 100s

[1] détaillée : time(1,2,3,4).google.com
[2] détaillée : a1nvlh0fc0asuq.iot.eu-west-1.amazonaws.com

l'ampoule LIFX, des communications périodiques ont lieu entre l'appareil et son propre serveur (hébergeur 2). Certaines de ces communications sont clairement utilisées comme mécanismes de Keep-Alive [4], mais d'autres paquets TLS peuvent être utilisé pour mettre à jour le statut de l'ampoule, bien que nous n'ayons aucun moyen de le prouver, le trafic étant chiffré sans solution pour accéder au texte en clair.

Nous avons constaté que la passerelle IKEA Tradfri ne génère aucun le trafic périodique vers Internet. Toutefois, il existe deux soi-disant intégrations pour cette passerelle IKEA : une pour Alexa et une pour Google Assistant, afin de permettent à ces assistants de contrôler l'ampoule. Si nous n'observons pas de transmission périodique lorsqu'aucune intégration n'est activée, nous en observons lorsqu'une intégration est activée, sans aucune différence entre Alexa et Google Assistant.

Si l'on considère l'ensemble des cas, sauf celui de la passerelle IKEA sans intégration, on observe que, mis à part les paquets NTP, il y a toujours un paquet TCP Keep-Alive et un paquet TLS envoyé périodiquement. Toutefois, l'ampoule LIFX envoie également un paquet TLS supplémentaire, ce qui en fait l'appareil qui envoie le plus grand volume de données de manière périodique : 188 octets par minute en moyenne. En comparaison, en moyenne, le pont Philips et la passerelle IKEA avec intégration envoient respectivement 48 et 174 octets par minute.

Le cas des scénarios d'usage

4. Philips a confirmé que ce trafic est causée par le mécanisme de Keep-Alive ou Heartbeat du protocole WebSocket (RFC 6455), voir Section 3.6.

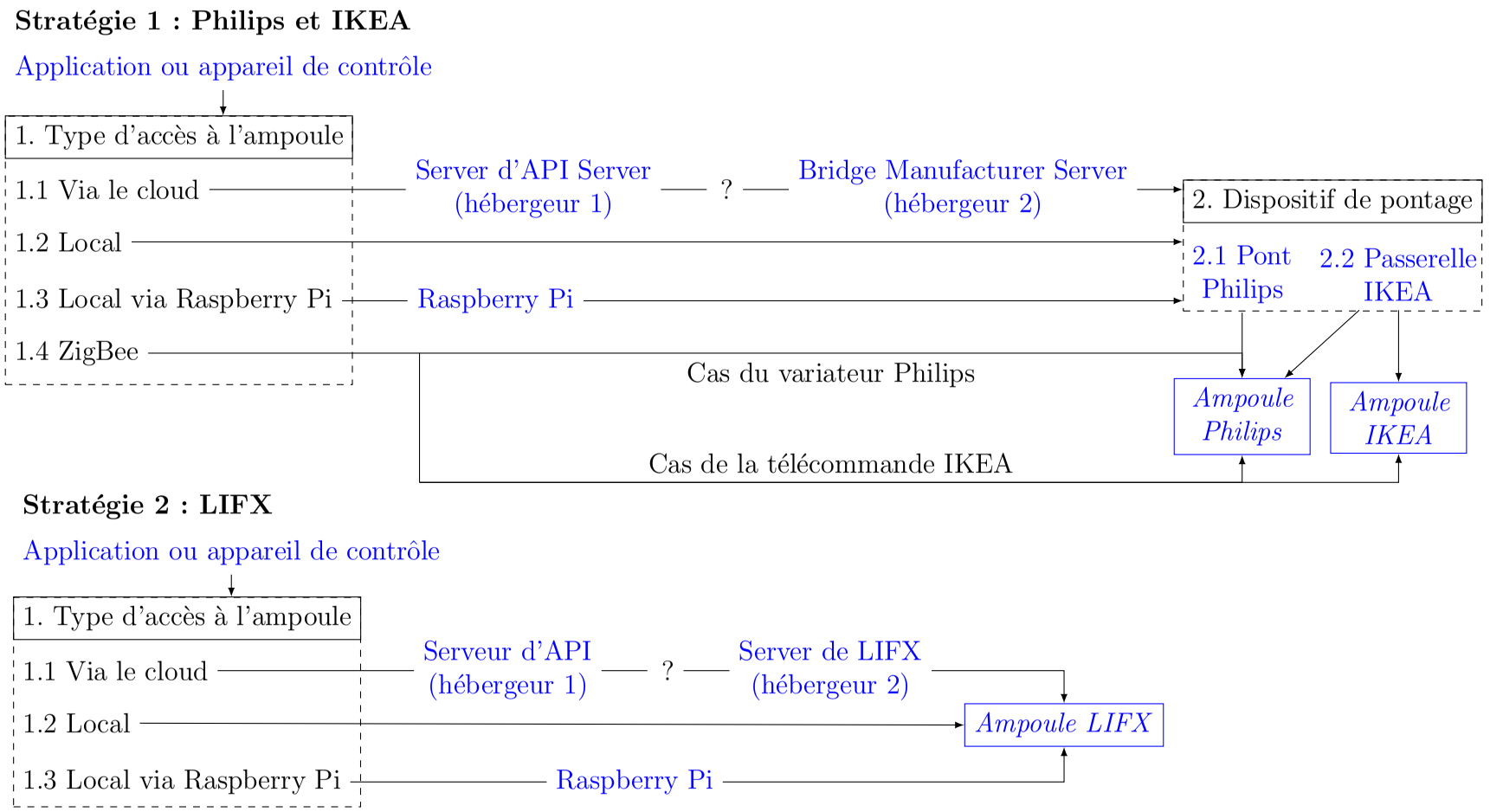


Figure 3.3: Taxonomie des chemins des requêtes ON/OFF pour les scénarios d'usage. *Nous identifions plusieurs catégories de chemins, représentatives des stratégies des fabricants, en fonction du type d'accès aux ampoules, du dispositif recevant la demande de l'application, et des ampoules. Les points d'interrogation font référence au trafic sur Internet qui nous est caché, le texte en bleu fait référence aux dispositifs physiques.*

Table 3.5: Hébergeur 1 et 2 selon l'application ou l'appareil de contrôle, le dispositif de pontage ou l'ampoule. Ce tableau complète la Figure 3.3.

Control Device or Application	Type d'accès à l'ampoule	Hébergeur 1	Peut utiliser : Pont Philips	Passerelle IKEA	Ampoule LIFX
Google Home	1	Google	✓	✓	✓
Google Home app	1	Google	✓	✓	✓
Amazon Echo Spot	1	Amazon	✓	✓	✓
Alexa app	1	Amazon	✓	✓	✓
IFTTT app	1	Amazon	✓	×	✓
Philips Hue app	2	n/a	✓	×	×
All4Hue app	2	n/a	✓	×	×
Hue Hello app	2	n/a	✓	×	×
IKEA Home smart app	2	n/a	✓	✓	×
LIFX app	2	n/a	×	×	✓
Home Assistant app	3	n/a	✓	✓	✓
openHAB app	3	n/a	✓	✓	✓
Variateur Philips	4	n/a	✓	×	×
Télécommande IKEA	4	n/a	✓	✓	×
Hébergeur 2 pour pont Philips :			Google		
Hébergeur 2 pour passerelle IKEA :			Amazon		
Hébergeur 2 pour ampoule LIFX :			Google		

Concentrons-nous maintenant sur les scénarios d'usage. Nous avons analysé les chemins de données pris lors des demandes ON/OFF. La figure 3.3 et le tableau 3.5 listent les quatre (Philips et IKEA) ou trois (LIFX) catégories observées :

— Catégorie 1, intitulée « 1.1 Via le cloud » dans la figure 3.3, concerne le cas général d'un chemin de données qui passe par Internet. C'est le cas des deux enceintes intelligentes (Google Home et Amazon Echo Spot), mais aussi des applications smartphone associées (Google Home et Alexa), ainsi que l'application IFTTT. Dans tous les cas, les données sont d'abord envoyées à un serveur d'API Web distant (par exemple, pour tu traitement automatique du language naturel (TALN), hébergé soit par Google soit par Amazon (appelé ci-après `hoster 1`). Ensuite, la requête associée est générée et atteint le serveur Philips, IKEA ou LIFX serveur distant, hébergé par Google ou Amazon (appelé ci-après `hébergeur 2`), mais cette partie de la communication ayant lieu sur Internet, les détails ne sont pas visibles. À partir d'ici, la demande est envoyée au domicile et atteint l'ampoule soit directement (LIFX), soit par un dispositif de pontage, à savoir le pont Philips ou la passerelle IKEA (cette dernière est également en mesure de contrôler l'ampoule Philips Hue).

— Catégorie 2, intitulée « 1.2 Local » dans la figure 3.3, concerne les applications pour smartphone qui peuvent communiquer avec l'ampoule sans passer par Internet lorsque l'utilisateur est chez lui. C'est le cas des applications Philips Hue, All4Hue, Hue Hello, IKEA Home smart et LIFX. Ici aussi, la demande est envoyée soit directement à l'ampoule (LIFX), soit par l'intermédiaire d'un dispositif de pontage (Philips et IKEA).

— Catégorie 3, intitulée « 1.3 Local via Raspberry Pi » dans la figure 3.3, concerne les systèmes openHAB et Home Assistant, où la requête atteint d'abord le serveur Raspberry Pi, qui l'envoie ensuite soit directement à l'ampoule (LIFX), soit par l'intermédiaire d'un dispositif de pontage (Philips et IKEA).

— Catégorie 4, intitulée « 1.4 ZigBee » dans la figure 3.3, concerne les deux boutons intelligents : le variateur Philips et la télécommande IKEA Tradfri. Ici, les boutons communiquent directement, via le réseau ZigBee, avec l'ampoule intelligente, en contournant tout autre équipement. C'est l'équivalent « connecté » d'un interrupteur classique d'ampoule. En raison du choix de la connectivité Wi-Fi pour LIFX, il n'existe pas d'équivalent pour leur l'ampoule, probablement en raison de la consommation d'énergie d'une connectivité Wi-Fi, incompatible avec un interrupteur sur batterie.

Le cas de l'utilisateur distant

Jusqu'à présent, nous avons examiné le cas d'un utilisateur à domicile. Nous avons également effectué des expériences sur la façon dont chaque technique de contrôle se comporterait si elle était utilisée depuis l'extérieur de la maison. Les deux enceintes intelligentes, de par leur conception, restent à la maison.

Dans le cas d'un smartphone à distance, la situation dépend de l'application utilisée. Les applications All4Hue et Hue Hello se connectent toutes deux directement au pont Philips, en utilisant le réseau domestique local. À notre connaissance, il n'existe aucune technique permettant de les utiliser de l'extérieur.

Il en va de même pour l'application intelligente IKEA Home, sauf qu'elle se connectera directement à la passerelle IKEA au lieu du pont Philips.

Sans surprise, toute technique de contrôle qui s'appuie sur le chemin de données « 1.1 Via le cloud » de la figure 3.3 peut être utilisée par un utilisateur distant. C'est le cas des applications Alexa, Home, IFTTT, Philips Hue et LIFX. Notez qu'une passerelle IKEA nécessite les intégrations Alexa ou Home Assistant avant de pouvoir être utilisée par les applications Amazon ou Google.

Le cas d'openHAB est intéressant car deux options sont possibles pour une commande à distance de l'ampoule. La première option consiste à avoir une redirection de port dans la box du FAI pour rediriger le trafic de ce port vers la Raspberry Pi locale. Un smartphone distant peut ainsi se connecter à la Raspberry Pi par le port ouvert et lui envoyer une requête. La deuxième option consiste à utiliser le service cloud myopenHAB, à laquelle l'application openHAB se connecte directement. Ce service cloud relaie la demande au Raspberry Pi, chez lui, grâce à une connexion permanente (un mécanisme de Keep-Alive est utilisé ici aussi pour maintenir cette connexion en cas d'utilisation du service cloud myopenHAB). Ce scénario est en fait un dérivé du chemin de données « 1.1 Via le cloud » de la figure 3.3.

Avec Home Assistant, la redirection de port est également possible. Comme Home Assistant ne fournit pas d'application, il utilise les « skills » d'Alexa ou de Google Assistant comme points d'entrée externes, avec une approche similaire à celle du chemin de données « 1.1 Via le cloud » de la Figure 3.3, avec le serveur d'API d'Amazon ou de Google comme hébergeur 1, le service cloud de Home Assistant comme hébergeur 2, et la Raspberry Pi comme relais interne.

Le fait de dépendre des services externes d'Amazon ou de Google peut créer des problèmes de vie privée. C'est pourquoi l'utilisation de la redirection de port sur la box du FAI de l'utilisateur, bien qu'un peu complexe à mettre en place, est intéressante du point de vue de la vie privée. Cependant, cela a des implications en matière de sécurité, car un port ouvert est également un point d'entrée potentiel pour un attaquant externe. C'est pourquoi le fait de s'appuyer sur des services cloud externes, fournis par des entreprises de confiance en matière de sécurité et de confidentialité (une hypothèse forte), peut être un compromis raisonnable. C'est le cas d'openHAB avec son service cloud myopenHAB. Si Home Assistant propose également son propre service cloud, il exige d'intégrer des

≪ skills ≫ externes d'Alexa ou de Google Assistant, ce qui est très discutable (l'utilisateur reste dépendant des services d'Amazon ou de Google).

3.3.2 Implications en matière de souveraineté

On observe l'hégémonie de Google et d'Amazon qui sont impliqués dans 28 scénarios sur 33. Vous pouvez vous référer aux figures 3.4 et 3.5 pour le détail des volumes sortants par destination. Ces figures détaillent la quantité de trafic envoyée vers les destinations distantes au cours des 33 scénarios d'usage, ainsi que l'hébergeur associé à ces destinations distantes. La première concerne les volumes sortants pour les scénarios à faible volumes sortants, n'impliquant pas de reconnaissance vocale, la deuxième concerne ceux à gros volumes sortants, impliquant de la reconnaissance vocale.

Cette situation est due aux services qu'ils fournissent, et aussi au fait que leur système d'hébergement (≪ hébergeur 2 ≫ en Figure 3.3) est très utilisé par les systèmes de Philips, IKEA et LIFX.

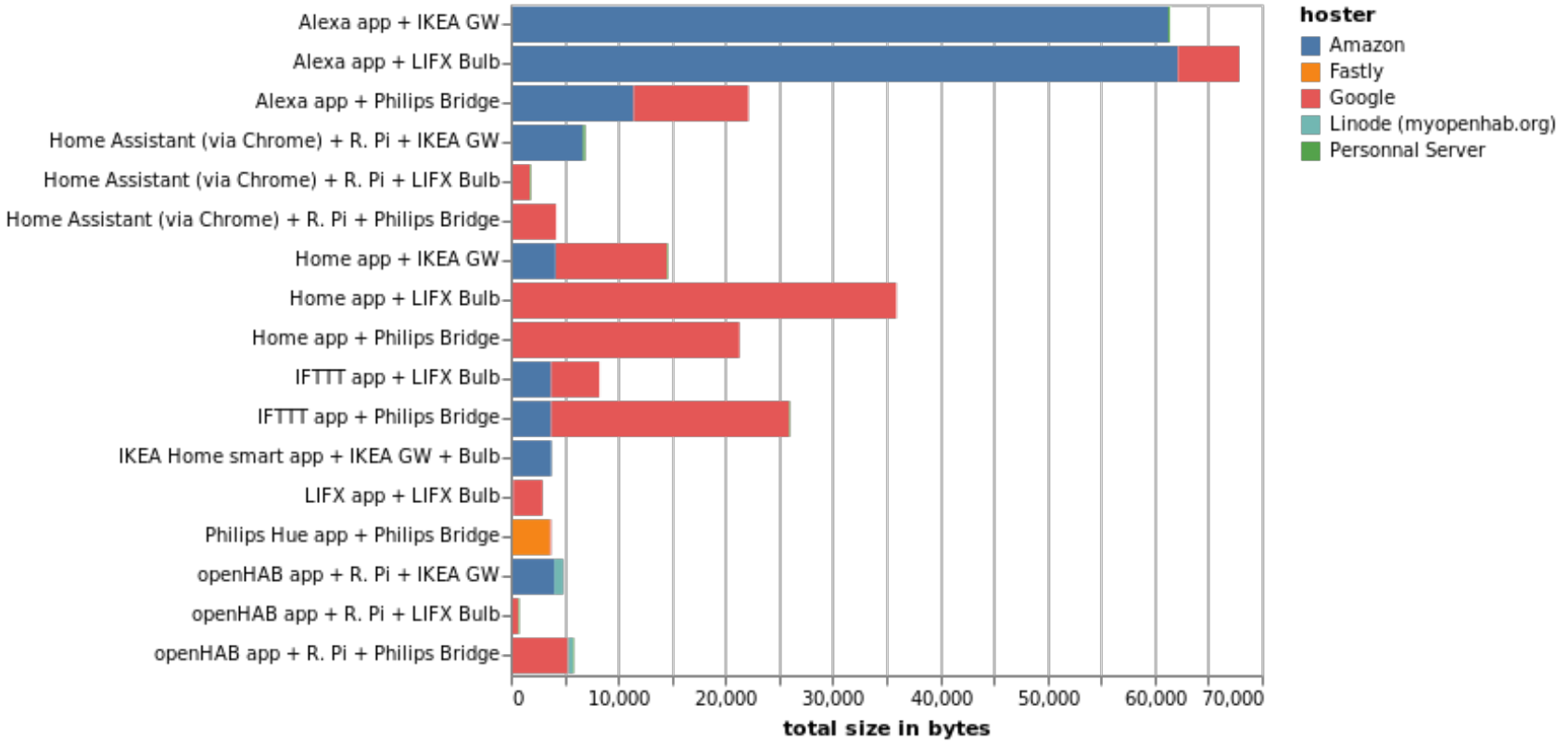


Figure 3.4: Volumes de données sortantes pour chacun des 21/33 scénarios d'utilisation sans microphone, pour une capture complète ON-OFF-ON-OFF, classés par destination. *Quatre scénarios d'usage ne sont pas présentés ici car ils n'envoient aucune donnée en dehors du réseau domestique : variateur + pont Philips, télécommande + passerelle IKEA Tradfri, application All4Hue + pont Philips et application Hue Hello + pont Philips. Cette figure diffère de la suivante par son échelle sur l'axe des x, puisqu'elle regroupe les scénarios où de* ***petites*** *quantités de données sont envoyées (celles qui n'impliquent pas de reconnaissance vocale).*

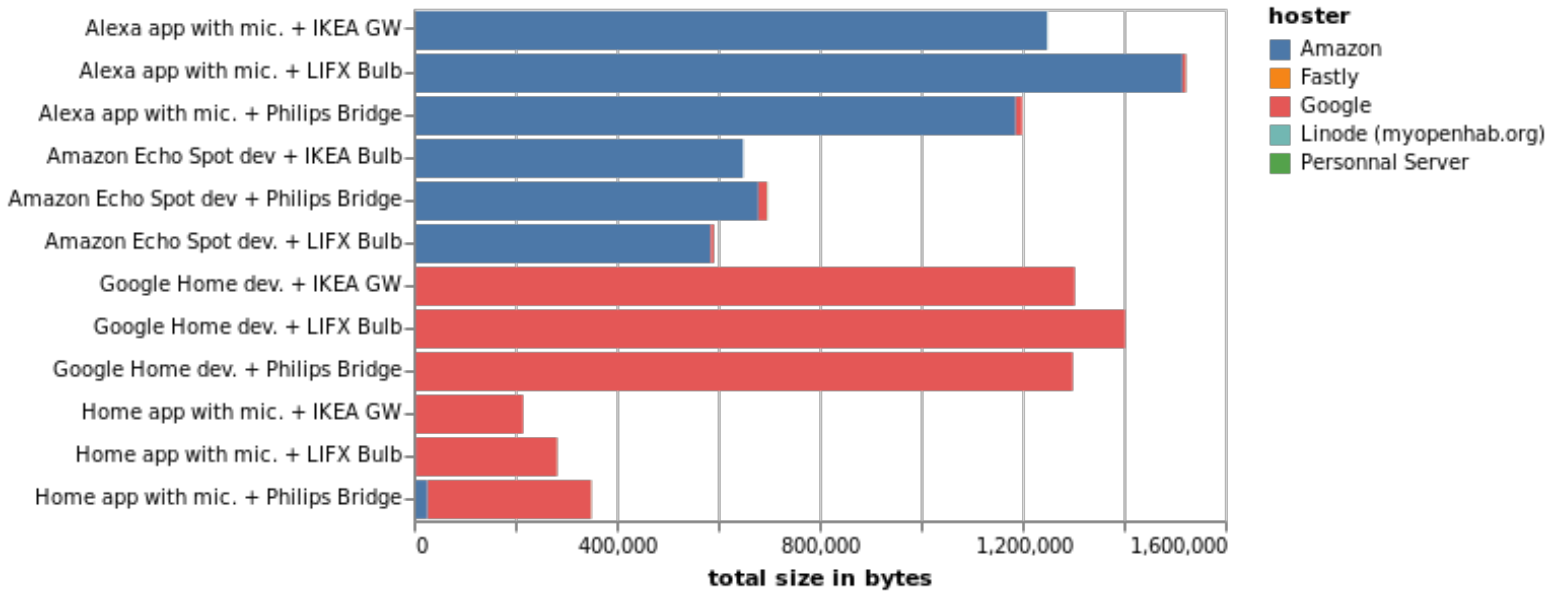


Figure 3.5: Volumes de données sortantes pour chacun des 12/33 scénarios d'utilisation avec microphone, pour une capture complète ON-OFF-ON-OFF, classés par destination. *Cette figure diffère de la précédente par son échelle sur l'axe des x, puisqu'elle regroupe les scénarios où de* ***grandes*** *quantités de données sont envoyées (généralement celles qui impliquent la reconnaissance vocale).*

On observe également que les scénarios impliquant un assistant vocal, que ce soit par le biais d'une enceinte intelligente ou d'une application pour smartphone, génèrent de 4 à 23 fois plus de données que le pire scénario sans reconnaissance vocale. Cela est dû au choix de Google et d'Amazon d'effectuer un traitement automatique du langage naturel (TALN) sur les serveurs de la société, ce qui est sans aucun doute efficace (puissance de traitement élevée lorsque nécessaire) et flexible (le TALN est facilement amélioré, sans nécessiter de mise à jour du côté client), mais aussi extrêmement coûteux en termes de communication (bien que nous ayons utilisé des commandes simples) [5].

Parce qu'il n'est pas possible actuellement d'obtenir une géolocalisation totalement fiable des serveurs distants, à partir de leur seule adresse IP, nous ne fournissons pas de résultats utilisant la géolocalisation des serveurs [6]. Cependant, il est clair (par exemple en analysant la RTT ou en utilisant l'outil traceroute) que plusieurs flux de données conduisent à des serveurs situés dans un pays éloigné, et parfois en dehors de l'Europe. Cette situation soulève également des questions de souveraineté, car indépendamment de la légalité de cette situation (la section 4.3 abordera cet aspect), elle crée une forte dépendance vis-à-vis des sociétés étrangères. Par exemple, une société américaine qui collecte des données auprès d'un citoyen européen est tenue de se conformer à une demande légale dans le cadre du Cloud Act.

Les transmissions périodiques, même si elles ne sont destinées qu'à maintenir la connexion en vie (ce qui est clair pour certaines d'entre elles, mais pas pour toutes), sont également préoccupantes. Elles donnent un accès permanent à l'appareil interne (pont Philips, passerelle IKEA ou ampoule LIFX) depuis l'extérieur de la maison, et parfois par des serveurs situés dans un pays non européen.

3.3.3 Inférence d'action utilisateur via l'analyse de la taille de la requêtes ON/OFF

Jusqu'à présent, nous avons examiné les chemins de données en général. Nous nous concentrons maintenant sur les messages eux-mêmes et commençons par montrer qu'une analyse de leur taille permet souvent de déduire l'action de l'utilisateur sur l'ampoule : allumage ou extinction. Ce type d'inférence correspond à plusieurs modèles d'attaquants :

1. Quelqu'un qui obtient un accès au réseau et surveille passivement le trafic.
2. Quelqu'un qui ne peut pas se connecter au réseau et qui surveille passivement le trafic.
3. Un FAI qui analyse les données envoyées vers ou depuis Internet. Cela limite les inférences relatives aux techniques de contrôle qui nécessitent l'utilisation de serveurs d'API externes.
4. Un dispositif malveillant qui surveille le trafic, conduisant à une situation similaire à la première.

Le tableau 3.6 montre que l'inférence d'action est souvent possible, même lorsque la requête est chiffrée. Ce tableau a été élaboré grâce à l'utilisation conjointe de l'outil `Burp` (avec trafic chiffré) et de la surveillance du trafic.

Tout scénario qui utilise le microphone est déclaré comme "n/a" (y compris les enceintes intelligentes) car la taille de la trame dépend fortement du trafic du microphone.

5. Cette situation soulève également des préoccupations en matière de durabilité.

6. Plus précisément, la localisation géographique des adresses IP recueillies dans le cadre de ces expériences était souvent fortement dépendante des outils utilisés. Bien qu'il soit clair que certains outils sont moins fiables que d'autres, nous n'avons pas réussi à tirer des conclusions non contestables et avons donc décidé d'omettre cette analyse. Cela peut aussi parfois être le résultat de services dynamiques, hébergés dans divers centres de données.

Table 3.6: Inférence d'action par taille de trame. *Un scénario dans lequel nous n'avons pas pu extraire une taille de trame cohérente est marqué "échec". Les scénarios impliquant des flux audio sont marqués "n/a" (aucune taille de trame cohérente). Les deux boutons intelligents sont ignorés (communications directes). Alexa est marqué « dyn. » parce que la taille de ses requêtes n'est pas constante mais peut toujours être utilisée pour déduire l'action. En rouge, on trouve les trames envoyées en clair.*

Trafic sortant :	Pont Philips		Passerelle IKEA		Ampoule LIFX	
	on	off	on	off	on	off
Amazon Echo Spot	n/a					
Google Home	n/a					
Alexa app	*dyn.*	*dyn.*	*dyn.*	*dyn.*	*dyn.*	*dyn.*
Alexa app avec mic	n/a					
Home app	245	246	245	246	245	246
Home app avec mic	n/a					
Hue Hello app	269	260	n/a			
IFTTT app	*échec*	*échec*	n/a		*échec*	*échec*
IKEA Home smart app	n/a		*échec*	*échec*	n/a	
Philips Hue app	259	260	n/a			
All4hue app	358	359	n/a			
Home Assistant (Chrome)	*échec*	*échec*	*échec*	*échec*	*échec*	*échec*
LIFX app	n/a				*échec*	*échec*
openHAB app	320	321	320	321	311	312
Hébergeur 2	639	640	*échec*	*échec*	*échec*	*échec*
R. Pi, openHAB	67	68	*échec*	*échec*	*échec*	*échec*
R. Pi, Home Assistant	85	86	*échec*	*échec*	*échec*	*échec*

Parmi les 33 scénarios listés dans le tableau 3.6, 12 requêtes ON et OFF pourraient être déduites des tailles des trames lors de l'examen des requêtes sortant de l'appareil ou de l'application. En outre, le pont Philips peut recevoir des requêtes en texte clair d'une Raspberry Pi, ou chiffrées à partir de l'hébergeur 2, mais dans les deux cas, la taille de la trame peut être utilisée pour déduire l'action. Ce n'est pas le cas avec la passerelle IKEA et l'ampoule LIFX. Nous pouvons voir que 11 captures utilisent des requêtes OFF qui ne diffèrent des demandes ON que d'un octet. Cette différence d'un octet s'explique facilement par l'utilisation des chaînes de caractères "ON" contre "OFF", ou "TRUE" contre "FALSE", ce que nous avons confirmé en examinant le texte en clair, grâce à `Burp`. L'analyse des trames d'IFTTT conduit à un échec car les requêtes ON et OFF font la même taille. Après avoir examiné le texte en clair, en utilisant `Burp`, nous avons découvert que cela vient du fait que la requête demande au serveur d'API de IFTTT de basculer l'état de l'ampoule en utilisant un identifiant de longueur fixe, et non pas de l'allumer ou l'éteindre. L'application IKEA Home smart est similaire, avec des requêtes ON et OFF de même taille. Toutefois, en raison de l'utilisation de DTLS, nous n'avons pas pu déchiffrer les messages et en comprendre la raison. L'application LIFX utilise son propre protocole où l'état de l'ampoule est représenté par un seul octet. L'application Alexa utilise des tailles dynamiques pour les requêtes ON et OFF, cependant les requêtes ON auront toujours une taille impaire et pour les requêtes OFF une taille paire.

Toutes ces inférences ont des implications en matière de protection de la vie privée :

savoir quand une ampoule est allumée ou éteinte permet de profiler très précisément l'activité d'un utilisateur. Bien entendu, un attaquant qui ne surveille que la taille des messages ne sera pas nécessairement en mesure de faire la distinction entre les deux types de requêtes (d'autres messages peuvent s'avérer être de la même taille), surtout si l'attaquant ne connaît pas la nature exacte de la technique de contrôle de l'ampoule. Toutefois, il s'agit d'une information qui peut aider à déduire l'action de l'utilisateur et, en tant que telle, elle contribue à la menace. En fait, la résolution du problème est triviale puisqu'il suffit d'avoir des requêtes de taille égale, en utilisant soit un rembourrage (padding), soit un encodage différent (par exemple, "1" ou "0").

3.3.4 Fuite de données dans le contenu des messages

Curieusement, lorsqu'on regarde le contenu des requêtes ON/OFF, celles-ci sont en majorité composées de données dédiés aux statut ON/OFF et aux rapports d'erreurs. À part le statut de l'ampoule, qui révèle des informations sur les habitudes des habitants, rien d'autre n'a été trouvé qui pourrait être considéré comme compromettant.

Table 3.7: Fuite de données dans les requêtes autres que ON/OFF (sélection de champs). *Ce tableau distingue les communications locales (en haut) et distantes (en bas). L'application Google Home est omise (les données ne correspondent pas à nos critères de sélection), ainsi que l'application IKEA Home smart (aucun accès au texte en clair).*

	De	À	Donnée
Local	Pont Philips	All4Hue app, Hue Hello app, openHAB and Home Assistant R. Pi	bridgeid, modelid, swversion, groups.*.name, lights.*.state.bri, lights.*.state.on, lights.*.swversion, lights.*.uniqueid
		All4Hue app, Hue Hello app, openHAB R. Pi	apiversion, gateway, ipaddress, name, whitelist.*.name, zigbeechannel, scenes.*.name
Remote	Alexa app	kinesis.us-east-1.amazonaws.com	app_id, version_name, client_id, country, device.make, device.model, platform.version, event_timestamp
		device-metrics-us-2.amazon.com	softwareVersion, model, CustomerId, OSType, androidId, DEVICE_MANUFACTURER, AppVersion, DEVICE_MODEL, OS_VERSION, EventTimestamp
		latinum-eu.amazon.com	latitudeInDegrees, longitudeInDegrees
	IFTTT app	api.segment.io	anonymousId, app.namespace, app.version, device.advertisingId, device.id, device.manufacturer, device.model, device.name, locale, os.version, userId, login, timestamp
		e.crashlytics.com	appBundleId, appVersionName, deviceModel, osVersion, timestamp
	LIFX app	30747.api.swrve.com	user, app_version, data.name (app.toogle_power), data.time, data.email
	cloud.lifx.com	LIFX app	product.name, product.identifier, brightness, power, room, firmware.version
	Philips Hue app	phillateemis.iad-03.braze.com	app_version, device_id, time, bridge_id, bridgeversion, bridgefirmware, user_id, roomdashboard_lights_switch.state
		api.amplitude.com	timestamp, user_id, device_id, uuid, version_name, os_name, os_version, device_brand, device_manufacturer, device_model, androidADID, bridge_id, timestamp, bridgefirmware, bridgeversion, event_id, event_type

La situation est différente en ce qui concerne les requêtes autres que ON/OFF. Le Tableau 3.7 liste quelques unes des données d'autres requêtes qui pourraient causer des problèmes en terme de vie privée et de sécurité. En particulier, on peut mentionner :

— Les identificateurs qui permettent le traçage de l'utilisateur ;
— L'Android Advertising Identifier (ADID), collecté par le traqueur Amplitude, inclus dans l'application Philips ;
— Des informations sur l'état de l'appareil pouvant être utilisées pour obtenir un aperçu de l'activité de l'utilisateur ;
— Géolocalisation avec une précision de 13 mètres ;
— Versions de programmes et firmware pouvant être utilisés pour trouver des exploits facilement, menant ainsi à des problèmes de sécurité ;
— La donnée du canal ZigBee peut informer un attaquant sur la façon de surveiller le réseau ZigBee ;
— L'adresse IP, le nom de l'appareil et le son modèle qui peuvent être utiles pour la découverte du réseau ;
— Les identifiants peuvent mener à des problèmes à la fois de vie privée et de sécurité, premièrement parce qu'ils peuvent renfermer des informations à caractère personnel et deuxièmement parce qu'ils peuvent mener à des authentifications interdites ;
— La liste blanche qui contient une liste de toutes les applications qui ont été utilisées par l'utilisateur pour interagir avec le pont Philips.

Nous constatons que certains messages révèlent plus que nécessaire : Amazon collectant des géolocalisations avec une précisions de 13 mètres est clairement disproportionné par rapport à la finalité, ce qui est interdit par le RGPD. Nous voyons également que certaines fuites sont dues aux traqueurs d'applications (c.-à-d., IFTTT, LIFX et Philips Hue), que nous aborderons dans la section suivante.

3.3.5 Les traqueurs d'applications

Les traqueurs d'applications pour smartphones jouent souvent un rôle important dans les fuites de données personnelles. Le tableau 3.8 montre la liste des traqueurs (connus) embarqués par les applications, qu'ils soient actifs ou non, identifiés à l'aide de Exodus Privacy [105]. Nous voyons ici encore que Google joue un rôle clé. L'application openHAB comprend deux traqueurs. Les applications All4Hue et IKEA Tradfri sont les seules pour lesquelles Exodus Privacy ne trouve pas de traqueurs. Nous reconnaissons que des solutions telles que Google CrashLytics de Firebase Analytics pourraient être extrêmement utiles aux développeurs pour améliorer la stabilité de l'application et comprendre comment les utilisateurs interagissent avec elle. Cependant, elles représentent néanmoins un problème de vie privée qui est très probablement mal compris, même par les développeurs eux-mêmes.

3.3.6 Caractéristiques bénéfiques pour la vie privée

Table 3.8: Traqueurs embarqués dans les applications smartphone.

Application	Traqueurs
Alexa	Amazon Advertisement
	Bugsnag
	Google CrashLytics
	Google Firebase Analytics
All4Hue	-
Home	Google Analytics
	Google Firebase Analytics
Home Assistant	n/a (no application)
Philips Hue	Amplitude
	Apptimize
	Braze
	Google CrashLytics
	Google Firebase Analytics
	HockeyApp
Hue Hello	Google CrashLytics
	Google Firebase Analytics
IFTTT	Google CrashLytics
	Google Firebase Analytics
	Segment
IKEA Tradfri	-
openHAB	Google CrashLytics
	Google Firebase Analytics

Table 3.9: Caractéristiques globalement bénéfiques pour la vie privée et situation des différentes techniques de contrôle. *La colonne « open-source » indique si l'application, le firmware ou le code source de la Raspberry Pi est accessible au public. Les deux colonnes suivantes indiquent si l'application ou l'appareil (y compris la Raspberry Pi si c'est sensé) utilise des communications chiffrées* par défaut*, localement ou à distance (si supporté). La colonne "messages à taille fixe" indique si l'application envoie des messages ON-OFF de taille fixe afin d'éviter l'inférence d'action (Section 3.3.3). Enfin, la colonne "fonctionne sans Internet" indique quelle technique de contrôle fonctionne encore en cas de coupure d'Internet.*

	Technique de contrôle	open-source	comm. locales chiffrées	comm. distantes chiffrées	messages à taille fixe	fonctionne sans Internet
appareil	Amazon Echo Spot	×	✓	✓	n/a	×
	Google Home	×	✓	✓	n/a	×
	Variateur Philips	n/a	n/a	n/a	n/a	✓
	Télécommande IKEA Tradfri	n/a	n/a	n/a	n/a	✓
application smartphone	Alexa	×	✓	✓	×	×
	All4Hue	×	×	n/a	×	✓
	Home	×	✓	✓	×	×
	Home Assistant + R. Pi	✓	×	×	×	✓
	Philips Hue	×	✓	✓	×	✓
	Hue Hello	×	×	n/a	×	✓
	IFTTT	×	✓	✓	✓	×
	LIFX	×	×	✓	✓	✓
	openHAB + R. Pi	✓	×	×	×	✓
	IKEA Tradfri	×	✓	n/a	✓	✓

Le tableau 3.9 résume plusieurs observations relatives à la vie privée faites dans le cadre de ce travail.

L'utilisation de solutions open-source est bénéfique pour la vie privée en raison de la transparence qu'elle offre : les fuites de données personnelles peuvent être identifiées, ainsi que l'adéquation du traitement des données, et un fork respectueux de la vie privée est également possible si la licence le permet. De ce point de vue, les projets open-source Home Assistant et openHAB sont les deux seules exceptions.

Nous constatons que le cryptage est ignoré par cinq techniques de contrôle lors des communications locales, ce qui signifie qu'elles reposent entièrement sur la sécurité Wi-Fi. Plus important encore, il est essentiel que tout message quittant le réseau domestique soit chiffré. Heureusement, la plupart des solutions utilisent des connexions HTTPS car elles reposent sur les serveurs externes d'API Web hébergés par Google ou Amazon, pour lesquels le chiffrement est obligatoire. Une exception existe avec les solutions open-source Home Assistant et openHAB qui utilisent HTTP *par défaut*, même s'il est possible de passer en HTTPS.

Cependant, comme nous l'avons vu dans la section 3.3.3, il est souvent possible de déduire une action de l'utilisateur en se basant sur l'analyse de la taille des messages, même avec le chiffrement. Le tableau 3.9 rappelle que les applications IFTTT, LIFX et IKEA Home smart sont robustes face à ce type d'attaque, en utilisant des messages de taille fixe pour contrôler les ampoules.

Enfin, être capable de contrôler une ampoule sans accès à Internet est à la fois résistant aux coupures de communications et bénéfique du point de vue de la souveraineté.

3.4 Rappel des travaux connexes et contributions

La littérature se concentre la plupart du temps sur un objet connecté à la fois, en essayant de trouver les vulnérabilités, les fuites de données ou les moyens de sécuriser les transmissions. Comme vu dans l'état de l'art plusieurs sujets ont été étudiés qui se rapprochent de nos travaux :

- identification des appareils et la reconnaissance des infrastructures sans fil [84, 85, 86, 172].
- déduction d'actions déclenchées par des appareils à partir des métadonnées des paquets ou de leur contenu en clair [90, 87, 110], ou de trames chiffrées [92, 93, 89].
- prévention des déductions d'actions [90, 109, 110].
- capture de trafic [173, 79] et statistique [107].
- identification de traqueurs et tierces parties [105, 106].

En revanche, notre travail se concentre sur les différentes manières de contrôler un dispositif donné, ce qui est également primordial. En effet, un appareil peut parfois être utilisé par plusieurs autres appareils. Dans ce but, le dispositif contrôlé exposera une API ayant des comportements différents selon la situation. Les travaux connexes qui concernent la protection de la vie privée se concentrent soit sur le smartphone, soit sur le point de vue de l'appareil. Nous pensons qu'il est important de combiner ces deux aspects, car les applications et les appareils influencent tous deux la manière dont les communications se déroulent. Bien que certaines de nos méthodes aient déjà été utilisées dans certains articles (par exemple, l'inférence d'action en utilisant les tailles de trame), c'était généralement pour le cas particulier d'un appareil cible donné et de son application officielle sur smartphone. Notre approche élargit l'analyse et met en évidence la complexité du monde réel : de nombreuses combinaisons (technique de contrôle, ampoule cible) permettent l'inférence d'action.

Le travail de Ren et al. [107] est le plus proche du mien mais diffère, comme vu en section 3.2, dans la méthodologie. La principale différence est que je m'intéresse à toutes

les techniques de contrôle. Mon travail se limite à un nombre restreint d'objets mais pousse l'étude sur chacun d'eux, de façon plus intrusive, tandis que Ren et al. ne font qu'observer, mais sur une grande quantité d'objets et à deux endroits différents et aux juridictions différentes : l'Angleterre et les États-Unis.

On retrouve des points communs entre les deux travaux : Google et Amazon reçoivent des informations sur quasiment tous les objets connectés, la plupart des données sont chiffrées, le chiffrement n'empêche pas les inférences et peu de données personnelles sont envoyées.

Cependant, du fait de la différence méthodolique, je peux confirmer ces faits, non seulement pour les intéractions entre les objets et Internet, mais aussi pour tout ce qui concerne les intéractions locales. J'ai également mis en place des attaques de l'homme du milieu, qui révèlent que même les échanges chiffrés ne partagent que peu de données à caractère presonnel. Une différence importante notée grâce à cette différence de méthodologie est que le chiffrement des communications en local ne représente qu'une faible majorité comparé au chiffrement des communications vers Internet.

Le contenu du trafic chiffré est souvent laissé de côté dans les travaux connexes, alors que nous avons utilisé plusieurs techniques pour examiner les trames chiffrées. Nous nous concentrons également davantage sur l'aspect de la souveraineté en examinant les destinations et les volumes de données, ainsi que les choix de conception des appareils et des applications pour smartphones en matière de protection de la vie privée.

3.5 Conclusion

Nous avons étudié comment l'utilisation de différentes techniques pour contrôler trois ampoules intelligentes représentatives et populaires fabriquées par Philips, IKEA et LIFX, peut changer radicalement les risques de sécurité et de vie privée. Nous avons examiné un large éventail de ces techniques : deux enceintes intelligentes, deux boutons intelligents qui contrôlent directement l'ampoule, dix applications pour smartphone (dont celles des fabricants d'ampoules) et deux systèmes de gestion de périphériques IoT opensources.

Nos travaux mettent en lumière plusieurs problèmes essentiels. Tout d'abord, un petit changement dans la manière dont l'utilisateur contrôle l'ampoule intelligente ciblée peut avoir des conséquences majeures à différents points de vue : risques de sécurité, fuites de données personnelles et souveraineté. Il n'est pas concevable pour un utilisateur d'anticiper de telles conséquences, car : (1) la tendance naturelle est de se concentrer sur l'appareil lui-même, et non sur la partie commande de l'appareil, alors que les deux sont essentielles et, (2) tout cela se cache derrière des interfaces utilisateur faciles à utiliser qui contribuent à faire croire à l'utilisateur qu'il a le contrôle total, alors que c'est exactement le contraire.

Deuxièmement, il met en évidence le rôle majeur joué par Google et Amazon dans les écosystèmes étudiés, ce qui n'est pas en soi quelque chose de nouveau. Ces acteurs sont au carrefour de nombreux flux, grâce à leurs enceintes intelligentes, à leurs propres services (par exemple pour le traitement du langage naturel), aux services fournis aux autres acteurs (par exemple, Google fournit ses services cloud à Philips et LIFX, Amazon à IKEA et IFTTT), et à leurs traqueurs d'applications. Ils sont en mesure d'en savoir beaucoup sur une maison intelligente et sur la vie de ses habitants.

Enfin, pour de nombreux acteurs, notre travail met en évidence des défauts de conception, leur solution n'étant pas conforme aux principes de respect de la vie privée dès la conception. Ne compter que sur les communications locales, lorsque cela est approprié (par exemple, lorsque l'utilisateur est à la maison, en utilisant un appareil connecté au Wi-Fi), devrait être la norme. Non seulement cela réduit les risques de fuites de données personnelles, mais cela rend la maison intelligente robuste face aux coupures Internet et cela contribue à prévenir les problèmes de souveraineté. La « minimisation des données » de-

vrait également être la norme. Nous ne voyons aucune justification réelle à la collecte de nombreux identifiants et d'autres métadonnées dans les messages échangés avec des serveurs distants par les applications IFTTT et Alexa, nous ne voyons aucune justification à l'inclusion de traqueurs commerciaux dans l'application Philips, car l'utilisateur a déjà payé pour cela (ce n'est pas un service gratuit), et Amazon collectant des géolocalisation avec une précision de 13 mètres est une absurdité total.

De ces deux points de vue, l'écosystème IKEA Tradfri, compatible avec les ampoules Philips et IKEA, est recommandé, à la condition expresse que l'utilisateur n'utilise pas le « skill » dédié qui permet une connexion aux services Amazon Alexa ou Google Assistant, ce qui ruine tout avantage en matière de protection de la vie privée. Le système Home Assistant, à la condition expresse que les « skills » Alexa ou Google Assistant ne soient pas utilisés (donc aucune communications distantes), est une bonne alternative pour les communications locales. Le système openHAB est également une bonne alternative qui favorise la vie privée par rapport à d'autres solutions, tout en permettant le contrôle à distance. La nature open-source de ces deux projets communautaires offre également une transparence totale, une caractéristique qui manque cruellement pour les produits commerciaux.

Nous demandons donc des architectures et des solutions différentes car, à l'exception d'IKEA, de Home Assistant et d'openHAB, la situation actuelle n'est ni respectueuse de la vie privée de l'utilisateur (beaucoup de choses se passent sans son consentement éclairé), ni souveraine (elle crée des dépendances à l'égard d'acteurs étrangers). Nous reconnaissons que s'appuyer sur des services centralisés, dans le nuage, est un choix architectural qui offre parfois des avantages techniques évidents (par exemple, une amélioration plus facile et continue des technologies de TALN) et ouvre la porte à des services évolués (par exemple, le croisement de données provenant de différents services, afin de proposer des services plus précis et plus significatifs à l'utilisateur final). Toutefois, ce choix a des conséquences majeures que nous avons examinées dans ce travail.

3.6 Divulgation responsable

Les auteurs de ce travail ont fait une divulgation responsable envers les différentes entreprises mentionnées dans cet article. La mise à disposition au public de ce document a été reportée de deux mois pour cette raison.

Le 12 août 2019, nous avons contacté toutes les entreprises pour lesquelles nous suspections ou avions trouvé un problème de sécurité ou de vie privée. Nous avons reçu des réponses d'All4Hue, d'openHAB et de Philips. En particulier, Philips a confirmé le mécanisme de Keep-Alive/Heartbeat que nous suspections, causé selon eux par le protocole WebSocket (RFC 6455) utilisé. Philips et openHAB ont été surpris par la présence de traqueurs. All4Hue et openHAB ont été surpris que nous soyons préoccupés par l'absence de chiffrement sur le réseau domestique. Tous étaient prêts à améliorer leurs services pour une meilleure protection de la vie privée. Philips a notamment mentionné qu'ils prévoyaient de passer progressivement du HTTP au « HTTPS uniquement » dans un avenir proche.

Nous avons reçu une réponse de Home Assistant, mais à ce jour, aucune suite n'a été donnée. Enfin, à ce jour, Amazon, Google, Hue Hello, IFTTT et IKEA n'ont jamais répondu.

Chapitre 4

Obtention de consentement : analyse des configurations et chartes de vie privée

La section 4.3 de ce chapitre est également tirée de l'article en cours de soumission dont la pré-publication se trouve sur HAL [147].

4.1 Introduction

Ce chapitre concerne la question du consentement des utilisateurs dans les applications liées aux objets connectés compris dans les habitats intelligents. Il vient complémenter l'étude technique du chapitre 3.

Le consentement est une notion intrinsèque à celle de vie privée et il parait primordial de réfléchir sur la façon dont celui-ci est recueilli. Dans le cas présent, deux angles sont adoptés pour juger de la probité des fabricants : les configurations que ce dernier permet à l'utilisateur et le contenu des chartes de vie privée. Les configurations (sur application ou en ligne) sont en effet la seule méthode disponible pour un utilisateur afin de gérer la façon dont ses données seront traitées par le fabricant, tandis que les chartes de vie privée établissent formellement les droits et devoirs que le fabricant a vis-à-vis de l'utilisateur.

Or comme je vais l'illustrer dans les prochaines sections, les possibilités de configurations des applications et objets connectés sont loin d'être adaptées au recueil de consentement. Dans certains cas il est même possible de ne pas donner son consentement du tout au fabricant et de quand même avoir accès au service qu'il fournit, alors que celui-ci peut nécessiter l'utilisation de données à caractère personnel.

Ensuite, nous ferrons un examen des chartes de vie privée fournies (ou non) par les différents fabricants et montrerons que la plupart d'entre elles n'informent pas l'utilisateur final de manière adéquate. Par conséquent, ces fabricants ne peuvent prétendre obtenir un consentement libre, éclairé, spécifique et univoque de l'utilisateur, ce qui rend la collecte de données à caractère personnel illégale selon le RGPD si l'on fait l'hypothèse que le consentement est la base légale, ce qui est réaliste ici.

Les applications et objets connectés dont les configurations et chartes de vie privée sons étudiées dans ce chapitre, sont hérités du chapitre précédent. Nous analysons les configurations des applications et objets connectés dans la section 4.2 et les chartes de vie privée dans la section 4.3. Nous conclurons en section 4.4.

4.2 Configuration des applications et objets connectés

Les fabricants, lors du développement de leurs applications, sont confrontés au choix de l'implémentation la plus efficace pour fournir leur service et la plus compatible avec leur modèle d'affaire. En étudiant les applications, objets et services que nous avons sélectionnés, on remarque que les fabricants optent la plupart du temps pour une solution qui permet à d'autres fabricants qu'eux d'accéder à leur propre écosystème. Pour ce faire, Philips, LIFX et IKEA utilisent une même méthode qui est d'exposer leur API aux autres services comme Google Home, Amazon Alexa ou IFTTT. Dans un premier temps, cela leur permet de développer leur propre application qui consiste finalement en une interface de leur API. Dans un second temps, cela permet à tout acteur désireux de supporter les objets connectés du fabricant, de pouvoir créer eux aussi une interface graphique liée à cette API. Ce faisant, les fabricants permettent à un plus grand nombre d'acteurs d'être compatibles avec leur service, et donc augmentent potentiellement leur notoriété et leur succès.

Or, comme nous l'avons déjà vu précédemment, de nouvelles techniques de contrôle amènent avec elles une multitude de problèmes. Le consentement est l'un d'eux. En effet, choisir une application plutôt qu'une autre peut empêcher l'utilisateur d'accéder à certaines fonctionnalités. Comme nous l'avons vu jusqu'ici, la sécurité et la vie privée ne semblent pas faire partie des priorités de certains fabricants lorsqu'il s'agit d'implémenter les fonctionnalités de leur API, et beaucoup d'efforts sont encore à faire. Ceci inclut le consentement, qui ne semble pas pris suffisamment au sérieux. Nous verrons dans cette section que l'obtention du consentement de l'utilisateur est une tâche bien difficile. Nous verrons que non seulement les applications développées par les fabricants ne sont pas adéquates concernant l'obtention d'un consentement libre, spécifique, éclairé et univoque, mais que les autres techniques de contrôle peuvent tout bonnement détourner les quelques mécanismes prévus par les applications officielles.

Dans toutes les situations que nous avons analysées, le consentement est recueilli lors de la première configuration de la technique de contrôle utilisée pour manipuler l'ampoule. C'est pourquoi nous nous focaliserons principalement sur cette partie. Je commencerai, en section 4.2.1 par une synthèse des problèmes rencontrés. Je poursuiverai, en section 4.2.2 par montrer que les applications officielles ne peuvent pas prétendre recueillir le consentement d'un utilisateur correctement. En section 4.2.3, je montrerai que les applications tierces ne peuvent pas le prétendre non plus, et que le consentement peut même être complètement omis. Je parlerai ensuite de ce que j'appellerai ci-après les *concentrateurs de services*, à savoir les techniques de contrôles qui permettent d'utiliser une multitude de service comme Google Home, Amazon Alexa et IFTTT. Je montrerai en section 4.2.4 que ces techniques de contrôles sont tributaires des services auxquelles elles sont liées. Je finirai sur la responsabilité des fabricants en section 4.2.5.

4.2.1 Synthèse des problèmes rencontrés

Je présente en figure 4.1 les différents problèmes rencontrés concernant les premières configurations des applications et l'obtention du consentement dans ces applications. Ces problèmes sont classés en fonction d'où ils sont rencontrés, à savoir :

Les applications officielles sont les applications développées par les fabricants des ampoules.

Les applications tierces sont les applications développées par d'autres développeurs, mais qui utilisent le même service (API) que les fabricant.

Les concentrateurs de services sont les applications développées par d'autres développeurs et qui ont pour but de concentrer les services (API) de plusieurs fabricants, et parfois ajoutent une couche d'intelligence supplémentaire pour faire interagir les services entre eux au moyen de règles automatiques.

On peut voir en figure 4.1 que la plupart des problèmes sont contenus dans les applications officielles à l'exception de l'omission totale du consentement qui est spécifique aux applications tierces et aux concentrateurs de services. Ceci est dû au fait que la plupart des problèmes liés au consentement sont présents à cause d'une mauvaise gestion de celui-ci de la part des fabricant, et plus particulièrement, de leur API. Or toutes les interfaces de ces API sont contraintes par elles. Donc les problèmes provenant de ces API seront répercutés sur les interfaces. On peut cependant accorder un point positif à ces applications, qui ne sera pas accordé aux concentrateurs de services (cas particulier d'openHAB) et aux applications tierces : elles n'omettent pas le consentement, ou du moins pas volontairement.

Les concentrateurs incluent la plupart de ces problèmes, mais en suppriment certains. Du fait que ce sont des concentrateurs, ils ne font que se connecter à l'API des services distants, et se reposent entièrement sur le mécanisme de connexion du service distant. Par conséquent, il n'y a plus de problèmes liés à l'ordre des instructions, ou à la validité du consentement en fonction du temps, ou d'interface trompeuse, ou de données supplémentaires obligatoires. Il existe aussi le cas particulier d'openHAB qui ajoute un problème par rapport aux applications officielles : l'omission totale du consentement utilisateur. En effet, openHAB se connecte localement aux API, et ne bénificient alors pas des mécanisme de connexion classiques des fabricants, qui eux demandent explicitement un consentement à l'utilisateur.

Enfin, les applications tierces n'incluent au final qu'une petite partie de ces problèmes. Cela est dû au fait que, comme les concentrateurs de services, ils ne doivent que se connecter à l'API et y donner accès à l'utilisateur, mais à l'inverse des concentrateurs de services (hormis openHAB), elles sont utilisables uniquement localement. Cette distinction fait que les mécanismes de connexion changent par rapport aux concentrateurs de services, les rapporchant du cas particulier d'openHAB. D'ailleurs, comme pour openHAB, ces applications tierces ont un gros défaut : elles omettent complètement le consentement utilisateur.

Les sections suivantes présentent ces problèmes dans les détails. Chaque section décrit les problèmes liés à un type d'interface spécifique, mais comme il y a des intersections entre chaque type d'interface, je présente les problèmes qui y sont rencontrés de manières flagrante, et les accompagne d'exemples.

4.2.2 Cas des trois applications officielles : un consentement fragile

Pour comprendre les défauts dont sont victimes les 3 applications officielles (Philips Hue, IKEA Tradfri, et LIFX) en matière de consentement, il faut analyser comment celui-ci est recueilli. Il s'avère que dans les applications que nous avons utilisées, le consentement est recueilli uniquement lors de la première installation et configuration de celles-ci. Cette section regroupe les différentes catégories de problèmes que nous avons découverts.

La figure 4.2 représente les différentes étapes rencontrées dans les configurations de chaque applications analysées. Nous y ferrons référence tout au long de cette section.

Catégorie 1 : Manque de durabilité du consentement

Le consentement, une fois recueilli, est vu comme éternel. Lors de notre analyse, nous avons constaté que le consentement était seulement demandé lors de la

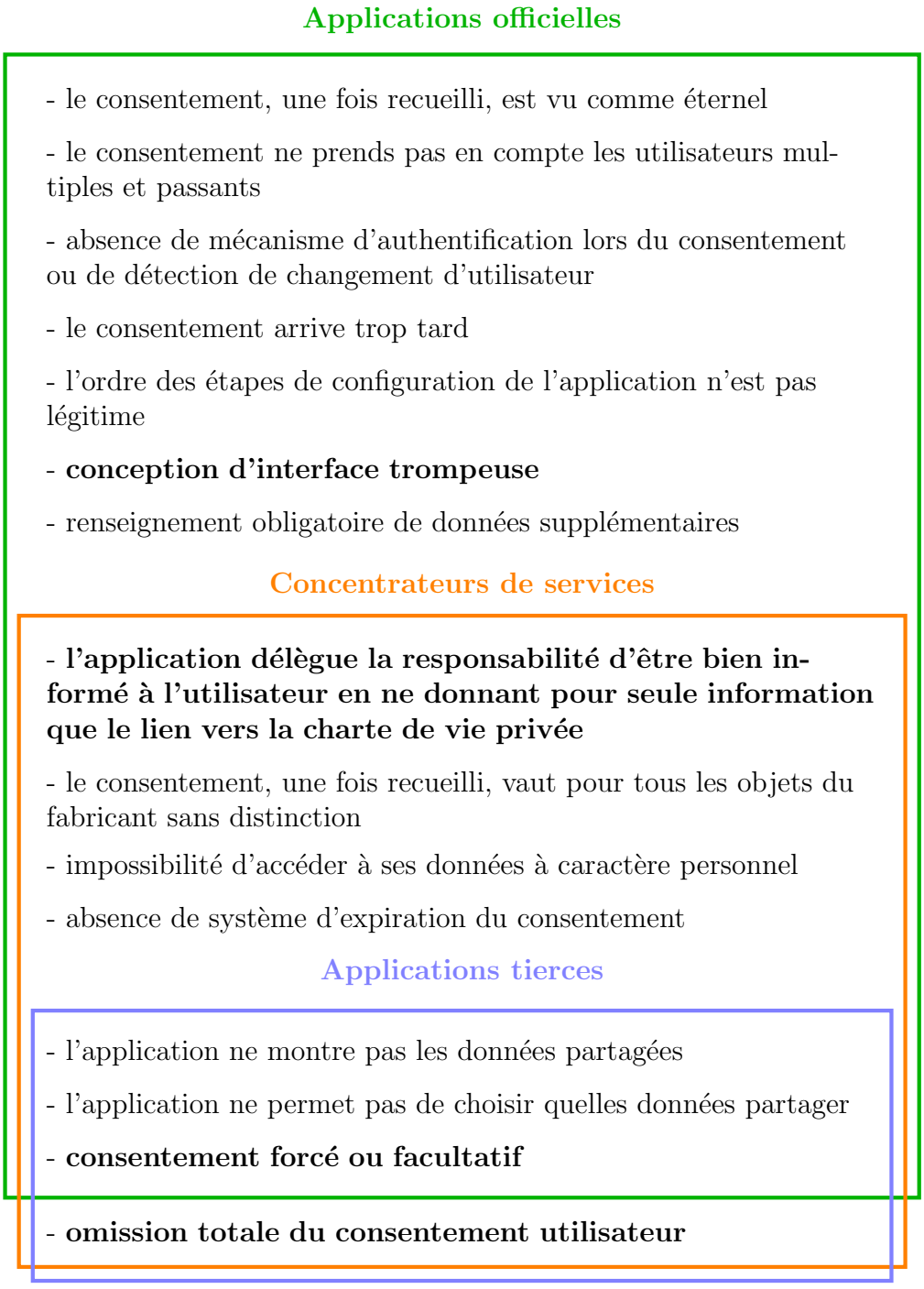


Figure 4.1: ***Synthèse des problèmes rencontrés liés aux premières configurations et au consentement en fonction des types d'interfaces utilisateur.*** *On constate que la totalité des problèmes rencontrés, à l'exception de l'omission totale de consentement, provient des applications officielles.*

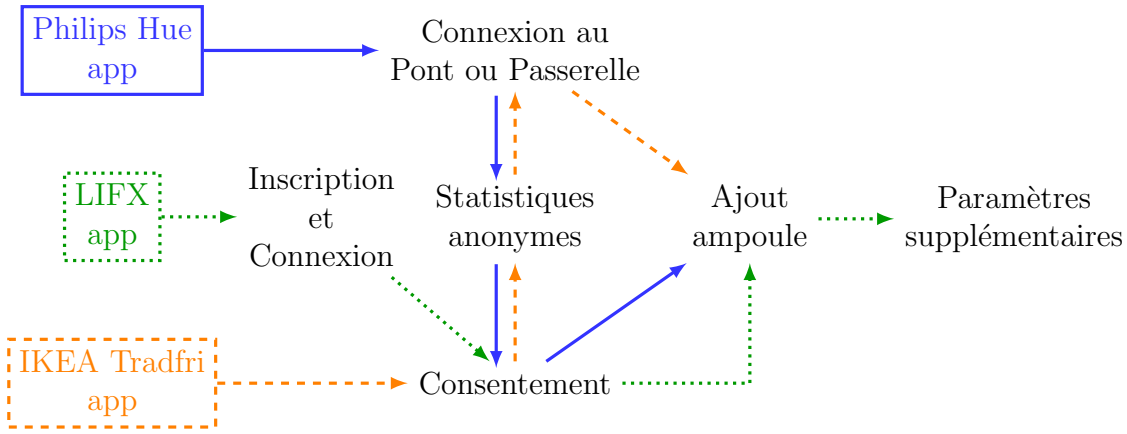


Figure 4.2: ***Étapes de la première configuration des applications officielles.*** *On peut noter que sur trois applications, on observe trois comportement différents. On peut remarquer que seule LIFX nécessite une étape d'inscription et des paramètres supplémentaires en fin de configuration. On voit aussi que Philips Hue et IKEA Tradfri ont un comportement quasi-identique, si ce n'est la différence d'ordre concernant le consentement, l'acceptation des statistiques anonymes et la connexion au dispositif de pontage.*

première configuration de l'application. Cela implique que le consentement n'est jamais remis en question par la suite, et donc, qu'il est éternel par défaut.

Ce problème a été relevé sur les trois applications officielles et peut avoir pour conséquence de rendre très difficile pour un utilisateur lambda de consulter son consentement, de le retirer, ou de le mettre à jour.

Le consentement ne prends pas en compte les utilisateurs multiples et passants. En effet, en observant les trois applications, on constate qu'aucune ne permet de protéger un utilisateur contre l'utilisation de ses données à caractère personnel lorsque l'appareil qui les contient a changé de mains. Autrement dit, un membre d'une famille propriétaire d'un objet connecté ne devrait pas avoir la capacité de récupérer des informations d'un autre membre de la même famille propriétaire du même objet, car ce dernier n'y a jamais consenti.

La gestion des comptes multi-utilisateurs peut être simplifiée en considérant que la famille est un seul et unique utilisateur, mais cela implique que tous les membres de la famille consentent à partager ouvertement toutes leurs données avec les autres membres. L'autre solution serait de séparer chaque utilisateur mais alors, s'ils ne sont pas d'accord sur ce à quoi ils consentent, cela peut créer des problèmes de fonctionnalités. Par exemple si une personne utilise une enceinte intelligente et qu'une autre passe à proximité alors qu'il n'a pas consenti à être écouté, alors le service sera momentanément coupé, ce qui coupera la première personne dans son activité.

S'ajoute ensuite le problème des utilisateurs passants, c'est à dire ceux qui n'utilisent pas un appareil mais qui, du fait de leur présence, partagent des données sans le savoir. C'est le cas d'un voisin qui parle à côté d'une enceinte intelligente. Il n'a pas déclaré son consentement à l'enceinte, et sa voix est capturée quand même.

Absence de mécanisme d'authentification lors du consentement ou de détection de changement d'utilisateur. Ce problème est lié au précèdent. En effet, comme on vient de le voir pour le cas de Philips, l'utilisateur peut consentir à partager ses données, mais il ne consent certainement pas à les partager avec un autre utilisateur du pont.

Pour éviter cela, la solution d'un compte ou au moins d'un mot de passe sur le pont, la passerelle, ou une ampoule pour chaque utilisateur peut permettre d'authentifier un utilisateur afin qu'il donne son consentement.

Ce genre de solution n'est cependant pas suffisante pour éviter la fuite de données. Comme montré avec l'exemple précédent sur l'enceinte intelligente, chaque appareil devrait être capable d'identifier qui est un utilisateur, et cela dépend énormément du contexte. Le simple fait d'être à moins de quelques mètres de l'enceinte fait de quelqu'un un utilisateur potentiel, qu'il fasse parti de la famille ou non.

L'implémentation d'un système de détection est alors endossée par le fabricant et pour chaque appareil indépendamment. Cependant, un système ouvert, commun, voire public, pour le consentement s'avérerait utile dans ce cas de figure.

Catégorie 2 : Pas de distinction de consentement en fonction des objets

Le consentement, une fois recueilli, vaut pour tous les objets du fabricant sans distinction. En effet, dans les trois applications analysées, le consentement n'étant demandé que lors de la première configuration, il est valable pour n'importe quel objet installé par l'utilisateur via l'application. Aucun consentement n'est redemandé lors de l'installation d'un nouvel objet.

On constate alors qu'il n'y a pas de distinction faite entre différents objets vis-à-vis du consentement. Une caméra est donc considérée équivalente en terme de consentement à une ampoule alors qu'on comprend facilement que cette dernière implique des mécanismes bien moins invasifs. Un utilisateur devrait d'ailleurs pouvoir consentir pour un objet mais ne pas consentir pour un autre, peut importe le moment où il l'installe. À notre connaissance, cela n'est pris en compte ni par l'interface utilisateur, ni par la charte de vie privée.

Catégorie 3 : Mauvaise chronologie des instructions

Le consentement arrive trop tard. Dans la figure 4.2, on peut voir que les trois applications utilisent un ordre différent. LIFX démarre par une inscription qui mène directement au consentement, tandis que Philips Hue et IKEA Tradfri ne requièrent pas d'inscription. Ces deux dernières applications ont un ordre très similaire, si ce n'est que tout ce qui se passe avant l'ajout d'une ampoule est inversé. En effet, IKEA Tradfri commence par le consentement, tandis que Philips Hue termine par celui-ci.

Philips Hue demande d'abord à l'utilisateur de connecter le pont à l'application, puis demande l'autorisation pour partager des statistiques anonymes afin d'aider à améliorer leurs services, et enfin demande le consentement. Peut-être que Philips avait une bonne raison de choisir cet ordre, mais le fait est que placer le consentement après ces étapes est un problème, puisqu'il est censé autoriser Philips à récupérer certaines données sur l'utilisateur. Or la simple connexion au pont peut déjà fournir théoriquement des informations sur l'utilisateur.

L'ordre des étapes de configuration de l'application n'est pas légitime. Concernant la première configuration de Philips Hue, nous ne pouvons pas être rassuré par l'ordre des étapes aurait d'ailleurs raison, puisque un accès au pont permet l'obtention des différentes applications utilisées auparavant par l'utilisateur précédent du pont, ou par lui même. L'ordre des étapes devrait donc être tant que possible irréprochable afin de donner confiance à l'utilisateur et d'éviter toute ambiguïté.

Catégorie 4 : Manque d'explicité des données partagées

L'application délègue la responsabilité d'être bien informé à l'utilisateur en ne donnant pour seule information que le lien vers la charte de vie privée. Les

trois applications obtiennent un consentement de la même manière, à savoir cocher une case pour stipuler qu'on est en accord avec les termes et conditions, et la charte de vie privée du service. La figure 4.3a montre la page affichée lors de cette étape. On peut y voir que seule une case à cocher est présente pour donner son consentement disant "I agree to the Terms & Privacy Policy". La figure 4.3b montre l'équivalent sur Philips Hue.

Il y a fort à parier qu'un utilisateur n'aura probablement ni la patience ni les compétences pour lire la charte de vie privée et pour comprendre quelles données à caractère personnel il devra partager, et a fortiori, les implications que cela peut avoir. Cela constitue en soi un gros problème puisque cela implique que l'utilisateur ne sera pas informé correctement, même si la charte de vie privée est bien rédigée. Même si certaines applications utilisent probablement ces systèmes de case unique à cocher par simplicité, le fait que ce genre de technique soit utilisé nous empêche de connaitre la nature des données partagées.

Catégorie 5 : Mauvaise interface graphique (Philips Hue uniquement)

Conception d'interface trompeuse. Si on observe la figure 4.3b, on peut remarquer que deux concepts sont mélangés dans l'interface : la demande d'autorisation à partager des statistiques anonymes pour l'amélioration du service et la validation des termes et condition, et de la charte de vie privée.

C'est un problème qui n'est pas anodin. Philips mélange deux concepts et engendre à l'occasion un problème plus grave encore car juste après ces instructions se trouve un bouton "enable". Or il est impossible de savoir à quoi correspond cette activation. Le petit lien en bas de la page indiquant "I do not want to help improve" n'arrange pas la situation, comme nous allons le voir maintenant.

Consentement forcé ou facultatif. En effet, si on regarde bien, la page d'aide anonyme possède une case pour le consentement, un bouton d'activation et un lien pour passer la page et donc ne pas aider Philips à améliorer ses services. Le problème, c'est que ce dernier lien peut être utilisé même sans avoir coché la case de consentement. Ce qui pose une nouvelle question : comment Philips gère le consentement depuis cette interface ?

Deux possibilités, toutes deux problématiques :

1. passer cette page correspond à une validation automatique de la case de consentement alors que l'utilisateur ne l'a pas réellement donné ;
2. passer cette page n'a pas d'effet sur le consentement et ce dernier est donc facultatif.

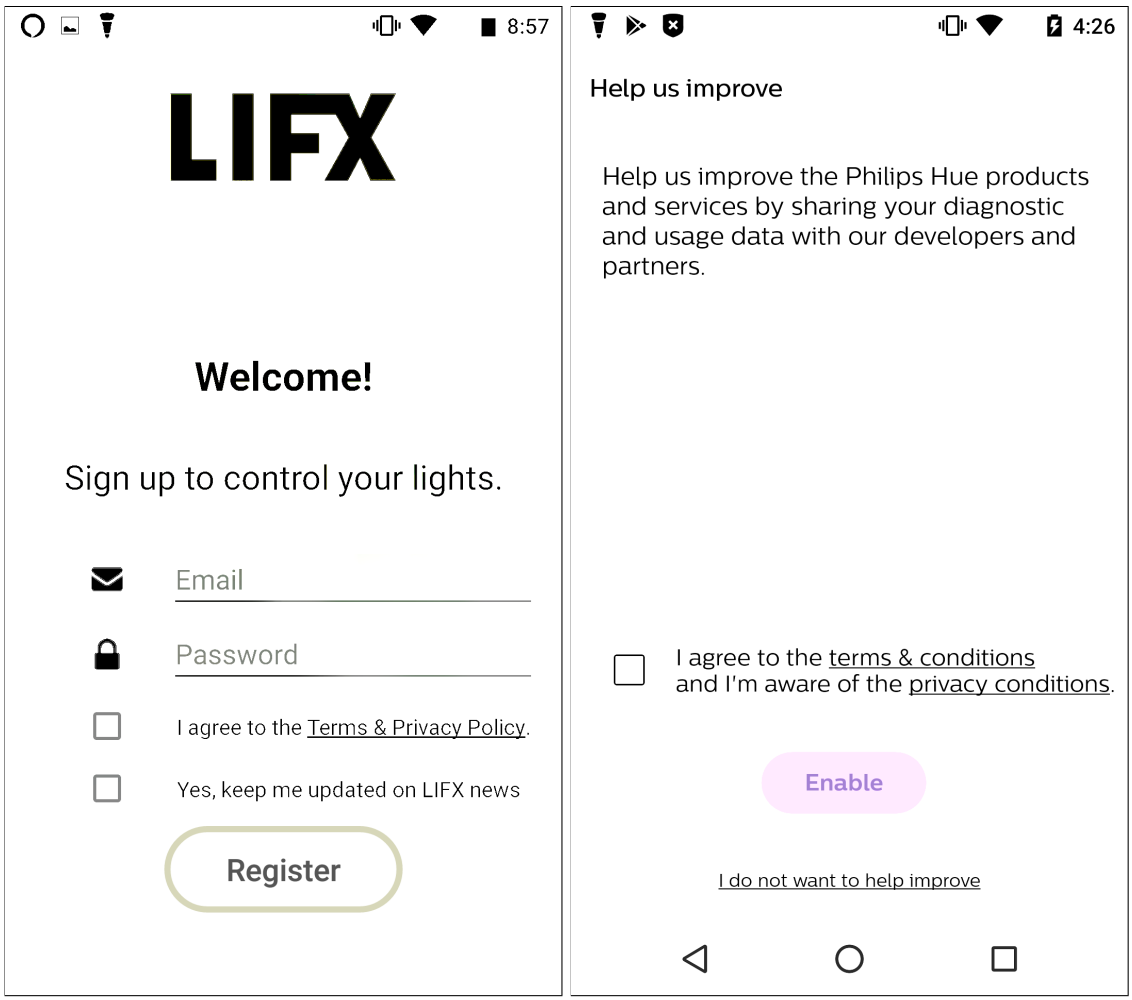
Catégorie 6 : Acquisition excessive de données

Renseignement obligatoire de données supplémentaires. Une dernière différence existe entre les configurations des trois applications : LIFX demande de donner la localisation de l'ampoule. Il n'est pas possible de ne pas en donner. Cela représente donc un dernier problème.

Une solution est possible, celle de ruser en créant une localisation factice à la place. Ainsi on satisfait la fonctionnalité sans donner d'information réelle sur l'emplacement de l'ampoule, ce qui est loin d'être une information anodine pour un cambrioleur par exemple, qui pourrait se servir de cette information pour connaitre la présence d'un utilisateur dans une pièce. Mais cette ruse n'est certainement pas à la portée d'un utilisateur lambda.

4.2.3 Des applications tierces qui prennent trop de libertés

Dans cette partie nous considérons les applications tierces qui visent à se substituer aux applications officielles, que ce soit complètement (copie conforme de l'application



(a) ***Interface de LIFX pour l'inscription et la connexion de l'utilisateur.*** *Une case concerne le consentement : "I agree to the Terms & Privacy Policy". LIFX délègue à l'utilisateur la responsabilité de sa compréhension de la charte de vie privée. Son consentement vaudra pour toute autre installation d'objet.*

(b) ***Interface de Philips Hue pour la demande d'aide via l'envoi de données et le consentement de l'utilisateur via l'acceptation des termes et conditions, et de la charte de vie privée.*** *Cette interface est un exemple de mauvaise conception graphique. On nous demande si on souhaite aider Philips, mais la case à cocher concerne le consentement, puis tout en bas, en petite police, on nous propose de ne pas aider. L'ordre est donc à revoir. De plus, un clic sur "I do not want to help improve" permet de passer à l'étape suivante de la configuration sans avoir coché la case de consentement au préalable.*

Figure 4.3: ***Captures d'écrans de LIFX et Philips Hue (versions éclaircies).***

officielle) ou partiellement (utilisation de quelques fonctionnalités particulières). Nous nous concentrons sur les applications listées dans le tableau 4.1.

Table 4.1: *Liste des applications tierces se substituant complètement ou partiellement (*) aux applications Philips Hue, IKEA Tradfri et LIFX.*

Application	Version	Remplace :
All4Hue	9.6	Philips Hue
Hue Essentials	1.7.0	Philips Hue et IKEA Tradfri
Chromania	1.11	LIFX*
Ma Deuce	1.6	LIFX*

Toutes ces applications ont plusieurs points communs :
— elles permettent d'accéder à des fonctionnalités fournies par les API qu'utilisent les applications officielles qu'elles remplacent ;
— elles ne nécessitent pas de compte et donc d'inscription et connexion ;
— elles ne demandent pas le consentement de l'utilisateur.
Ce dernier point constitue un problème majeur.

Omission totale du consentement utilisateur. L'état initial de l'API devrait par défaut refuser tout accès aux fonctions qui concernent des données personnelles. Un consentement devrait être nécessaire spécifiquement pour l'utilisation de ce type de fonction.

Or, comme on peut le constater, les applications tierces dépassent les problèmes des applications officielles il n'est même plus question de comment l'utilisateur peut consentir puisque le consentement est tout simplement inexistant.

Le fait d'utiliser une application tierce ne devrait pas permettre de se passer du consentement utilisateur. Le fait que cela soit possible dans chacune des applications tierces analysées ici prouve que la conception de l'API de chaque service correspondant, à savoir celles de Philips Hue, IKEA et LIFX, ne prends pas du tout en compte le consentement. Du moins si ce dernier est pris en compte, il n'est pas nécessaire au fonctionnement de leur écosystème, c'est-à-dire que toutes les fonctions de l'API sont disponibles, peu importe si elles ont été consenties ou non. C'est pourtant un problème qui peut être résolu très facilement en ne permettant l'accès à une fonctionnalité que si celle-ci a fait l'objet d'un consentement préalable. Évidemment, s'il y a beaucoup de fonctionnalités, comme le consentement est explicite pour une finalité donnée, faire des groupes de fonctionnalités interdépendantes serait plus pertinent. Aussi, dans le pire des cas, au moins avoir un consentement maître qui, s'il n'est pas donné, empêche tout simplement d'accéder aux fonctionnalités traitant des données à caractère personnel.

Ce problème est d'autant plus grave sur les applications All4Hue et Hue Essentials qui permettent, comme les applications officielles Phlips Hue et IKEA Tradfri, de connecter un pont Philips ou une passerelle IKEA, puis d'ajouter une ampoule. Ces applications utilisent les même mécanismes que les applications officielles. Pour Philips, cela correspond à une connexion au pont avec un clic sur le bouton du pont pour valider la connexion, puis à l'ajout des ampoules reconnues par le pont. Pour IKEA, cela correspond à une connexion à la passerelle grâce au flashcode collé dessus, ou au code de sécurité fourni à côté du flashcode, puis l'ajout d'une télécommande reconnue par la passerelle, et l'ajout d'une ampoule reconnue par la passerelle et située à proximité de la télécommande.

Pour ce qui est des applications Chromania [174] et Ma Deuce [175], l'impact est moins grand, non pas parce que LIFX a mieux implémenté son API, mais parce que ces deux applications n'en utilisent pas toute l'étendue. En effet, ces applications ne permettent pas de se connecter à un compte LIFX, mais tirent avantage de la connectivité Wi-Fi de l'ampoule. Pour qu'une application puisse accéder à une ampoule LIFX, il faut avant toute

chose que l'ampoule soit configurée pour être connectée automatiquement au réseau Wi-Fi de l'utilisateur. Cette configuration se fait sur l'application officielle, après la période d'inscription et de connexion au compte LIFX. Nous n'avons pas connaissance d'application tierce capable de se substituer complètement aux mécanismes d'inscription, connexion et configuration de l'ampoule, mais il n'est pas impensable qu'une telle application existe.

Même si Chromania et Ma Deuce n'ont pas la capacité de gérer un compte LIFX, ces deux applications peuvent tout de même manipuler les ampoules pourvu qu'elles soient connectées au même réseau que le smartphone de l'utilisateur. Cela est possible grâce au protocole de LIFX qui permet l'envoi de commandes LIFX en local. Consentement à part, cela ouvre également la porte à certaines attaques où l'attaquant se trouverait dans le même réseau que l'utilisateur. On peut imaginer des attaques similaires aux attaques pouvant créer des crises d'épilepsie [96] ou les attaques de canaux auxiliaires [94] déjà vues dans le chapitre 2.

4.2.4 Concentrateurs de services

Nous avons vu les problèmes pregnants concernant les applications officielles et les applications tierces, mais il existe un autre type d'interface capable d'intéragir avec les API de certains fabricants, dont Philips, IKEA et LIFX.

J'appellerai ces interfaces des concentrateurs de services puisqu'ils ont pour but principal d'être connectés à d'autres services, bien qu'ils permettent aussi d'ajouter un niveau supplémentaire d'interaction. Par exemple, certains comme openHAB permettent de créer des règles automatiques relativement poussées qui permettent d'utiliser le statut d'un appareil ou d'un service, ou de recevoir une instruction pour déclencher des actions. Par exemple, on peut déclencher l'allumage d'une ampoule lorsqu'on reçoit une requête « Allume l'ampoule » envoyée par Snips d'après une commande vocale d'un utilisateur.

Cette section traitera des problèmes rencontrés en particulier chez ces concentrateurs de services, à savoir les problèmes de transfert de consentement, le manque de granularité dans les données partagées et le manque de transparence.

Catégorie 1 : Transfert du consentement

Omission du consentement. Le premier concentrateur auquel nous allons nous intéresser est openHAB. Le but d'openHAB est d'avoir une interface commune pour différents services. Chaque service est géré par un add-on particulier qui se connecter à l'API du service en arrière plan, et donner accès graphiquement à ses fonctionnalités à l'utilisateur. Cela permet à l'utilisateur d'avoir une vue précise des capacités du service.

En revanche, tout comme pour les applications tierces, les add-on d'openHAB qui concernent les API de Philips Hue, IKEA Tradfri et LIFX, ne demandent jamais de consentement explicite de l'utilisateur. Ceci est encore une fois dû au fait que les API fabricant ne rendent pas obligatoire ce consentement.

En ce sens, on peut dire qu'openHAB échoue à transferer le consentement de l'API à l'utilisateur, mais qu'en amont l'API devrait forcer ce consentement pour éviter de déléguer cette responsabilité à ses différentes interfaces (applications tierces et concentrateurs de services inclus).

Catégorie 2 : Manque d'explicité et de granularité dans les données partagées

L'application ne montre pas les données partagées. Ceci concerne Google Home, Amazon Alexa et IFTTT. Les trois applications passent par une API externe de Philips,

IKEA et LIFX. En effet, bien que les objets connectés associés (Google Home et Amazon Echo) restent par définition à l'intérieur de la maison, les applications, elles, doivent pouvoir accéder aux objets des fabricants depuis n'importe où. Or les API externes sont accessibles à distance et localement. Google, Amazon et IFTTT cherchent donc à obtenir un accès distant quoi qu'il arrive aux services qu'ils supportent.

Les fonctionnements des trois services Philips, IKEA et LIFX concernant l'accès aux fonctionnalités à distance sont présentés en figure 4.4. Ils sont très similaires, à l'exception d'IKEA. Lorsqu'un programme souhaite accéder à un de ces services, il devra présenter à l'utilisateur une page web du fabricant lui demandant de se connecter à son compte, puis affichera une page concernant les données qui seront partagées et si l'utilisateur consent à ce partage. IKEA fonctionne dans l'autre sens : c'est leur application qui active une intégration vers Google ou Amazon.

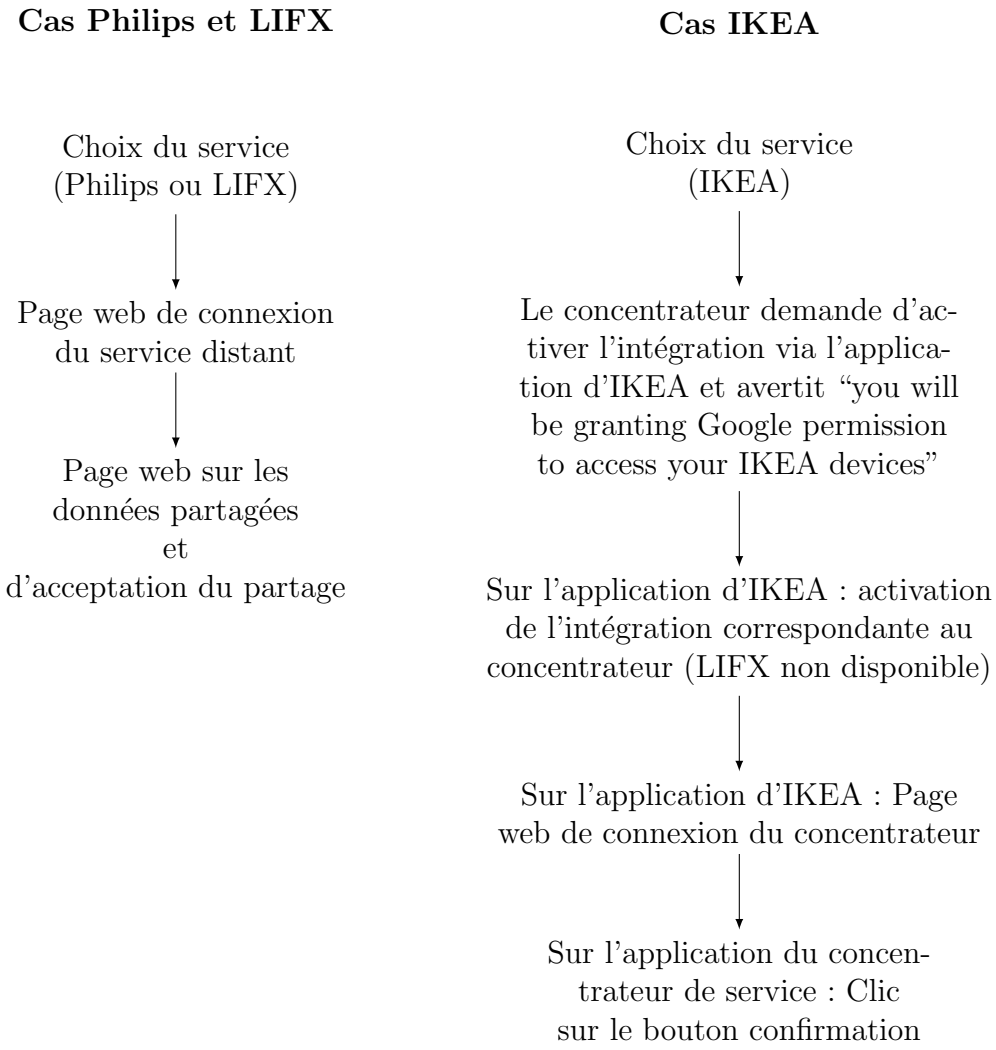
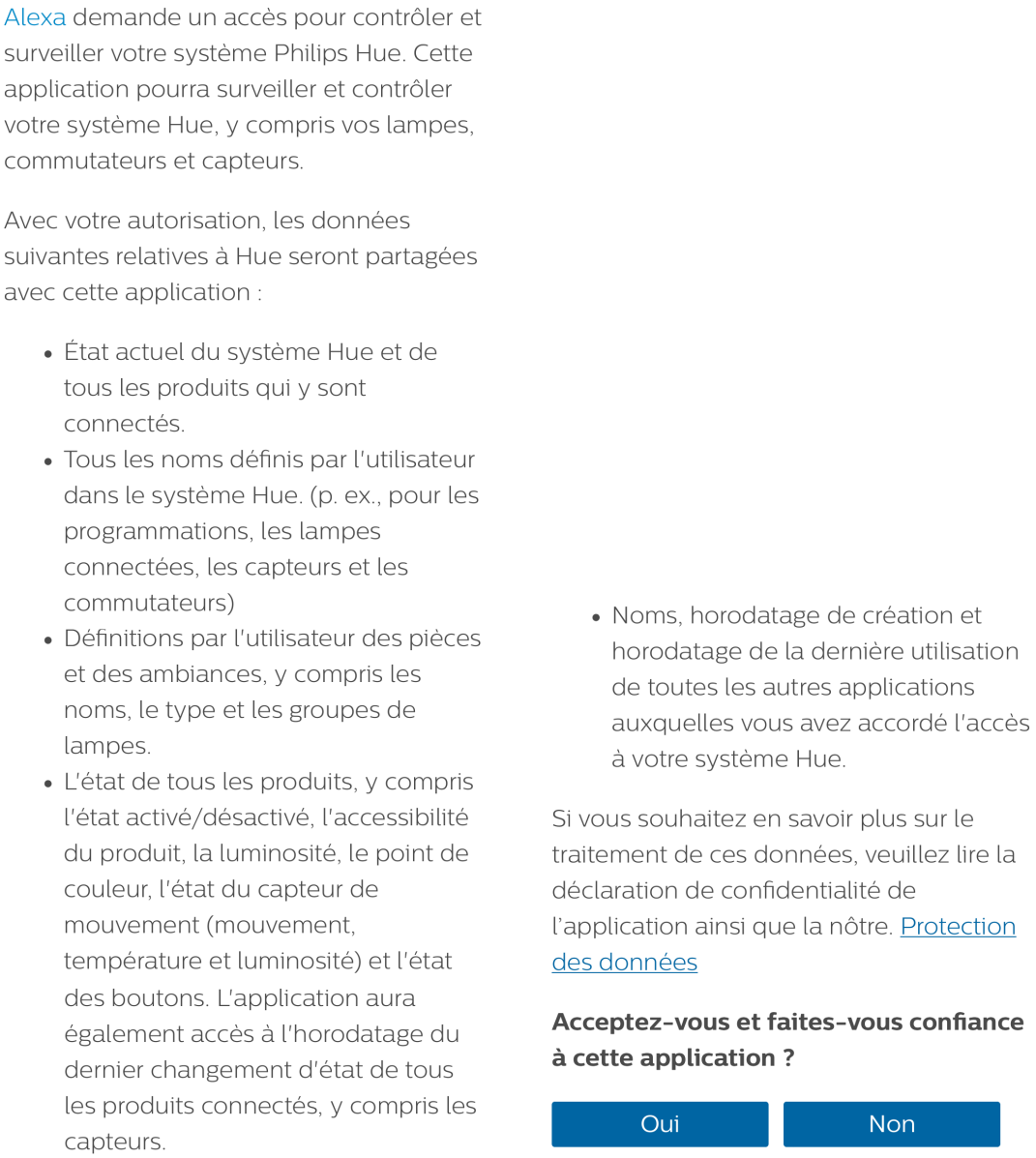


Figure 4.4: ***Étapes du consentement utilisateur via un concentrateur de service (Google Home, Amazon Alexa et IFTTT).*** *Nous ne mentionnons pas openHAB ici puisqu'il se connecte localement aux ampoules et ne demande pas de consentement.*

En ce qui concerne le partage des données, Philips, IKEA et LIFX ne sont pas égaux. Là où Philips précise relativement bien les types de données partagées (voir figure 4.5), IKEA et LIFX ne font qu'informer l'utilisateur que Google, Amazon ou IFTTT sera en mesure de contrôler ses appareils, ce qui est un manque d'explicité flagrant.



(a) ***Première partie.*** **(b)** ***Seconde partie.***

Figure 4.5: ***Page affichée par Philips lors de l'autorisation d'accéder à son API externe via Google Home, Amazon Alexa ou IFTTT.*** *Alexa est le cas présenté ici, mais le mot Alexa est remplacé par le nom des autres concentrateurs dans les autres cas. On peut remarquer que Philips a le soucis du détail ici, ce qui n'est pas le cas d'IKEA et LIFX qui ne donnent pas ce genre d'information.*

L'application ne permet pas de choisir quelles données partager. Dans le cas de Philips, le niveau de détail en ce qui concerne la liste des données partagées est très bon. Cependant, il serait bénéfique pour l'utilisateur d'être en capacité de choisir explicitement lesquelles de ces données il accepte de partager. Par exemple, s'il veut juste être en mesure d'éteindre ses lampes à distance, Google n'a pas besoin de connaitre quoi que ce soit sur son écosystème Philips, si ce n'est l'identifiant d'une règle Philips hébergée sur le pont de l'utilisateur. Lui permettre d'accéder à l'état des ampoules serait à la limite justifiable, mais les noms des pièces, ambiances, groupes, les luminosités, couleurs, etc., n'est pas pertinent.

Catégorie 3 : Manque de transparence

Absence de système d'expiration du consentement. C'est un argument discutable, mais le consentement d'un utilisateur ne devrait pas être considéré comme éternel, comme nous l'avons déjà vu précédement. Une date d'expiration du consentement devrait être donnée formellement à l'utilisateur, date après laquelle le service se doit de réobtenir son consentement. Cela peut être fait grâce à une notification par courriel, ou plus simplement via l'application elle même puisqu'on suppose que si l'utilisateur n'a plus l'application (et donc pas de notification), il n'utilise plus le service, et son consentement est donc expiré.

Cela est aussi vrai en particulier lorsque une nouvelle fonctionnalité ou un nouveau type de donnée est capturé, notamment par un des service connectés à un concentrateur de service. Une notification indiquant la différence de version devrait être envoyée à l'utilisateur dès lors que cette nouvelle version change de comportement vis-à-vis de ses données et sa vie privée, avec idéalement un historique des changements apportés.

L'ajout de système de règles automatique mentionné précédement (déclencher une action automatiquement à partir d'une autre) est aussi un problème lorsqu'il s'agit d'expiration de consentement, puisque ces règles ont pour but de fonctionner même sans utiliser l'application (hormis lors de la première configuration des règles). Cela veut dire que ces règles continuent à fonctionner tant que l'utilisateur ne les désactive pas explicitement. Or dans le cas des changements de versions du concentrateur de service ou du service lui-même, on ne peut pas confirmer que le concentement de l'utilisateur est toujours valide, et les règles automatiques doivent par conséquent être intérrompues.

Impossibilité d'accéder à ses données à caractère personnel. Depuis l'arrivée du RGPD, un utilisateur a le droit de faire une demande à un fabricant pour récupérer l'intégralité de ses données personnelles. Nous avons fais l'expérience avec Google et Philips, et avons remarqué que ce genre de manipulation n'est toujours pas au point. Nous voulions récupérer les informations personnelles obtenues par Google et par Philips par rapport à l'interaction entre Google Home et Philips Hue grâce, d'un côté, à Google Takeout, et d'un autre côté, à la page de demande de récupération de données de Philips [1]. Malheureusement, ces pages n'ont rien retourné, ce qui laisse à penser que la notion de données à caractère personel est certainement mal comprise par ces fabricants. Si on suppose que ces services sont honnête, on peut aussi supposer qu'il ne peuvent pas donner l'historique des allumage et exctinctions des ampoules, puisqu'ils les auraient supprimés, n'en ayant pas besoin. Cependant quelques informations comme les logins, l'activation du mode "Out of home", les noms des ampoules ou des scènes devraient être retournées, ce qui n'est pas le cas.

1. Page disponible ici.

4.2.5 Responsabilité des fabricants

Au final, même si les applications officielles, les applications tierces et les concentrateurs de services peuvent tous faire plus d'efforts pour corriger les problèmes énumérés dans les sections précédentes, les fabricants ont une responsabilité encore plus forte : gérer le consentement convenablement afin que ces applications ne puissent pas commettre ce genre d'erreur.

En effet, qu'un utilisateur puisse utiliser les fonctionnalités d'un service alors que ces fonctionnalités traitent des données à caractère personnel et que son consentement n'a jamais été donné explicitement est inadmissible. Il est nécessaire de créer un mécanisme qui puisse demander et valider le consentement utilisateur avant de donner l'accès aux fonctionnalités du service demandant un partage de données à caractère personnel.

Il est également de leur responsabilité de renseigner le plus efficacement possible l'utilisateur, et donc d'adapter ses interfaces pour lui faciliter la tâche. Il n'est pas raisonnable de déléguer à l'utilisateur la responsabilité de bien s'informer en lisant une charte de vie privée aride.

Même si cela est moins primordial, permettre une meilleur granularité dans le choix des données qu'on accepte de partager devrait également être pris sérieusement en considération.

4.3 Chartes de vie privée et consentement

Nous avons traité le problème de la collecte de consentement dans les applications d'objets connectés et leur premières configurations. Mais le consentement est aussi le point central des chartes de vie privée. Il est important de bien analyser les ambiguïtés qui y sont présentes puisque c'est aussi ce document qui doit permettre à l'utilisateur de donner un consentement valide.

Dans cette section, nous allons montrer que le consentement, du fait de certains défauts dans les chartes de vie privée, même s'il est recueilli, ne peut pas être concidéré comme libre, spécifique, éclairé et univoque.

4.3.1 Le besoin d'un consentement utilisateur

Nous considérons un utilisateur final vivant en France qui a acheté des appareils connectés, dans des magasins français et a téléchargé des applications depuis la version française de Google Play (PlayStore). Les sections précédentes ont montré que les acteurs dont les appareils et les applications ont été pris en compte manipulent des données personnelles (en effet, les identifiants techniques et les UUID sont collectés ainsi que des données brutes). Ces acteurs sont donc des responsables du traitement des données (s'ils "déterminent les finalités et les moyens du traitement") ou des sous-traitants (s'ils "traitent des données à caractère personnel pour le compte du responsable du traitement") dans le cadre du RGPD [176, art. 4(7), 4(8)]. Il s'ensuit qu'une base juridique valide est nécessaire. Dans notre cas, le RGPD exige un consentement libre, spécifique, informé et non ambigu de la personne concernée (par souci de simplicité, nous utiliserons le terme "utilisateur") [176, art. 4(11)].

4.3.2 Au sujet des chartes de vie privée

Toutefois, les informations fournies et le consentement obtenu soulèvent plusieurs questions de conformité réglementaire, résumées dans le tableau 4.2 et détaillées ci-dessous.

Le consentement n'est pas informé. Pour que les utilisateurs soient considérés comme légalement informés du traitement de leurs données personnelles, le responsable

Table 4.2: Problèmes rencontrés en matière d'information et de consentement. *Le temps de lecture est estimé en considérant une vitesse de lecture de 250 mots par minute, généralement acceptée pour les textes anglais (nous avons supposé qu'elle était valable pour le français également). Ce temps ne tient pas compte des spécificités des politiques de protection de la vie privée et constitue une estimation approximative à prendre avec précaution. Le symbole (*) indique un problème d'accessibilité.*

		consentement informé	consentement libre	Nombre de mots	Temps de lecture moyen
Ampoules	Philips	✓	✓	1615	6'
	IKEA	× (*)	✓	1369	5'
	LIFX	×	×	4814	19'
Appareils	Amazon	×	×	4981	20'
	Google	×	×	6009	24'
Applications smartphone	All4Hue	✓ (*)	✓	164	< 1'
	Hue Hello	✓ (*)	✓	808	3'
	IFTTT	×	×	5522	22'
	openHAB	×	✓	2335	9'
	H. A.	×	✓	1206	5'

du traitement doit (1) fournir aux utilisateurs certaines informations obligatoires, et (2) rendre ces informations "concises, transparentes, intelligibles et facilement accessibles", en utilisant un "langage clair et simple" [177].

Le consentement n'est pas éclairé parce que, premièrement, certaines chartes de vie privée ne sont tout simplement pas accessibles sur le site web du contrôleur des données, mais exigent de se rendre sur un service tiers (par exemple, Google Play) pour trouver le lien. Par exemple, il n'y a pas d'accès direct aux chartes de vie privée des applications d'IKEA, Hue Hello et All4Hue sur leurs sites web respectifs.

Deuxièmement, même lorsque les chartes de vie privée sont facilement accessibles, elles sont trop longues pour être lues par les utilisateurs. Par exemple, la version française de la charte de vie privée de Google [178] est une page HTML de 9 800 mots, ce qui correspond à 21 pages imprimées (la version PDF fournie par Google atteint même 33 pages). Les politiques Amazon [179] ou LIFX [180] sont des pages web de plus de 4 800 mots, ce qui correspond à 11 pages de texte. La politique de confidentialité d'IFTTT [181] est affichée sur une seule page web, au-dessus des conditions d'utilisation, en anglais uniquement (nous ne pouvons pas supposer que notre utilisateur, qui vit en France, comprend l'anglais). La page web complète, y compris la charte, représente un total de 10 000 mots, soit 20 pages imprimées, la charte elle-même représentant 11 pages du total [2]. Cela confirme les résultats d'études antérieures concernant la longueur des politiques et le temps nécessaire pour les lire [182]. L'utilisateur doit également lire les informations présentées. La plupart des utilisateurs ne lisent pas les politiques selon la littérature existante [183].

Pour que les utilisateurs donnent leur consentement éclairé, les informations fournies par le responsable du traitement doivent également être compréhensibles. Toutefois, la

2. Les comparaisons de pages HTML sont effectuées en utilisant la page principale de la charte, sans liens vers des références externes, en considérant un format A4, une police Times New Roman et une taille de 12pt lorsque nous indiquons un nombre de pages.

majorité des chartes de vie privée sont rédigées dans un langage peu accessible, surtout pour les utilisateurs profanes, et emploient un jargon juridique et une formulation ambiguë. Par exemple, la charte d'IFTTT fait largement usage de mots comme « may » ou « some », dont l'utilisation est explicitement découragée par les lignes directrices du WP29 en matière de transparence [177, sec 13.]. Certaines chartes de vie privée regroupent différents services d'un même fournisseur sur une même page, ou sont éparpillées sur plusieurs pages, voire sur plusieurs domaines. Par exemple, la charte de vie privée de Google contient de nombreux liens relatifs aux chartes des autres services du groupe, ce qui rend difficile pour un utilisateur d'avoir une vue d'ensemble claire du traitement global des données. À ce sujet, en janvier 2019, la CNIL a prononcé une sanction de 50 millions d'euros à l'encontre de cette société pour "manque de transparence, information insatisfaisante et absence de consentement valable" [184].

Le consentement n'est pas libre. Enfin, pour que le consentement soit considéré comme valide, il doit être donné librement. Toutefois, lorsque l'acceptation d'un traitement de données est obligatoire pour accéder à un service, le consentement ne peut pas être considéré comme libre. Par exemple, la dernière version de la charte de vie privée de LIFX le précise explicitement : "Si vous n'êtes pas d'accord avec nos politiques et pratiques, sauf indication contraire, vous avez le choix de ne pas utiliser ou accéder aux services".

Lorsque la politique ne bloque pas explicitement les utilisateurs qui refusent le traitement de leurs données, la charte peut être rédigée de manière à favoriser l'acceptation du traitement, en maximisant les avantages et en omettant de mentionner les risques. Par exemple, la page d'IFTTT contenant la charte de vie privée et les ToS commence par une déclaration commerciale intitulée : « Trust ¿ GDPR : Building trust takes more than just legal compliance ». Cette pratique, connue sous le nom de « framing » (ou cadrage) [185], peut inciter l'utilisateur à partager plus de données que prévu. La réglementation ne l'interdit pas explicitement, mais les lignes directrices du WP29 indiquent que le consentement peut être rendu nul par "tout élément de pression ou d'influence inappropriée" sur l'utilisateur [186].

Il semble peu probable que les utilisateurs de ces services puissent trouver, lire et comprendre la plupart de ces différentes chartes de vie privée. Il semble encore moins probable que ces mêmes utilisateurs soient en mesure de donner un consentement "libre, spécifique, informé et univoques" comme l'exige le RGPD. Toutefois, l'absence de consentement rendrait le traitement des données personnelles illégal, à moins qu'une autre base juridique ne s'applique.

En outre, lors du dernier examen annuel en novembre 2019, le Contrôleur européen de la protection des données (CEPD) a déclaré qu'il avait encore "un certain nombre de préoccupations importantes" concernant le Privacy Shield, dans le cas des sociétés basées aux États-Unis. Dans ses conclusions, son rapport a souligné "l'absence de contrôles substantiels" sur les aspects commerciaux, l'application complexe des principes concernant les transferts ultérieurs et les sous-traitants de données, le manque de transparence concernant la collecte de données par les autorités publiques, et les pouvoirs insuffisants du médiateur [187, sec. 112 – 115].

Pour ces raisons, nous pensons que la plupart des responsables du traitement des données ne peuvent prétendre obtenir un consentement valide, bien qu'il faille faire la différence entre les initiatives communautaires à code source ouvert et les entreprises commerciales. Cela s'applique également à une société basée aux États-Unis, qu'elle participe ou non au programme du Privacy Shield : la collecte, le traitement et le transfert de données personnelles d'un utilisateur de l'UE (notre cas) devraient être considérés comme illégaux en vertu de la GDPR [3]

3. Cette affirmation reflète l'opinion des auteurs et doit être confirmée par les agences européennes de protection des données et par un tribunal.

4.4 Proposition de solution

Nous avons vu dans ce chapitre que même si les fabricants prétendent recueillir un consentement, dans les cas étudiés, celui-ci n'est soit pas recueilli du tout, soit pas convenablement.

L'API développée par le fabricant ne semble pas prendre en compte le consentement correctement, et l'accès à ses fonctionnalités dans ces conditions devrait être, sinon impossible, du moins restreint, afin d'éviter à l'utilisateur de partager des données contre son gré. Un gros effort doit être fourni sur la gestion du consentement utilisateur concernant non seulement le fait de le donner, mais aussi de le retirer ou de le mettre à jour. Une plus grande granularité devrait aussi être permise dès la conception des API pour éviter toute ambiguïté qui pourrait être engendrée, par exemple, par l'introduction de nouveaux objets ou fonctionnalités demandant de nouvelles données.

Améliorer ces API aura également pour bénéfice de restreindre les possibilités d'envenimer la situation lorsqu'une application officielle ou tierce souhaite l'utiliser. En effet, dans ce chapitre nous avons vu que la plupart des problèmes se trouvent dans les applications, mais qu'ils sont permis *in fine* par l'API. Si l'API gère correctement le consentement, il sera plus difficile pour les applications de faire des erreurs vis-à-vis de celui-ci.

Bien informer l'utilisateur est d'ailleurs un problématique à part entière. Les informations apportées dans les applications des fabricants sont insuffisantes, voire parfois inexistantes. Les utilisateurs sont donc contraints de se replier sur la charte de vie privée du fabricant. Or, la plupart du temps, aucune information n'est réellement accessible, les informations fournies par les responsables de traitement des données dans leurs chartes de vie privée étant très discutables.

Avec toutes ces remarques, on constate qu'un outil commun entre les fabricants, les utilisateurs et les chercheurs en vie privée pour la gestion du consentement serait pertinent. La figure 4.6 donne un exemple de solution simple qui tire en partie son inspiration des travaux de Victor Morel sur le consentement dans l'Internet des objets [188].

Dans sa thèse, M. Morel relève certains problèmes vis-à-vis des solitions de partage de consentement existants, dont principalement le manque d'interface facile à utiliser et le caractère trop spécifique de ces solutions, incompatible avec l'hétérogénéité de l'Internet des objets.

Il propose en conséquence une solution de partage de consentement, soit direct (en échangeant avec l'appareil lui-même), soit indirect (en échangeant avec un registre accessible sur Internet).

Ma solution utilise une architecture très similaire, à ceci près qu'elle échange avec l'API du fabricant de l'objet, que ce soit directement (API de l'appareil ou de la passerelle) ou indirectement (API en ligne).

En revanche, le but de ma solution est différent. Il est de fournir un consentement sur les données brutes de l'utilisateur, mais aussi sur les données déductibles à partir de celles-ci. Cela implique l'existence d'une banque d'inférence qui serait un service dédié à la recherche d'inférences liées à une donnée. Ainsi l'utilisateur peut être informé des risques qu'il prend en partageant une donnée et, si risque il y a, il pourra refuser de la partager.

Là encore se trouve une autre différence, car le but de ma solution est d'empêcher une API de collecter les données interdites par l'utilisateur. En effet, une fois l'API renseignée du consentement de l'utilisateur, celle-ci ne devra plus exposer les fonctionnalités requierant ces données. On s'assure ainsi que le fabricant ne pourra pas collecter la donnée.

Un outil qui permettrait de lister les données à caractère personnel collectées et l'étendue des inférences pouvant être faites grâce à ces données et, si besoin est, permettrait à l'utilisateur de bloquer le partage des données posant problème.

Cette figure montre le principe des « tables de consentements ». Je vais détailler ci-

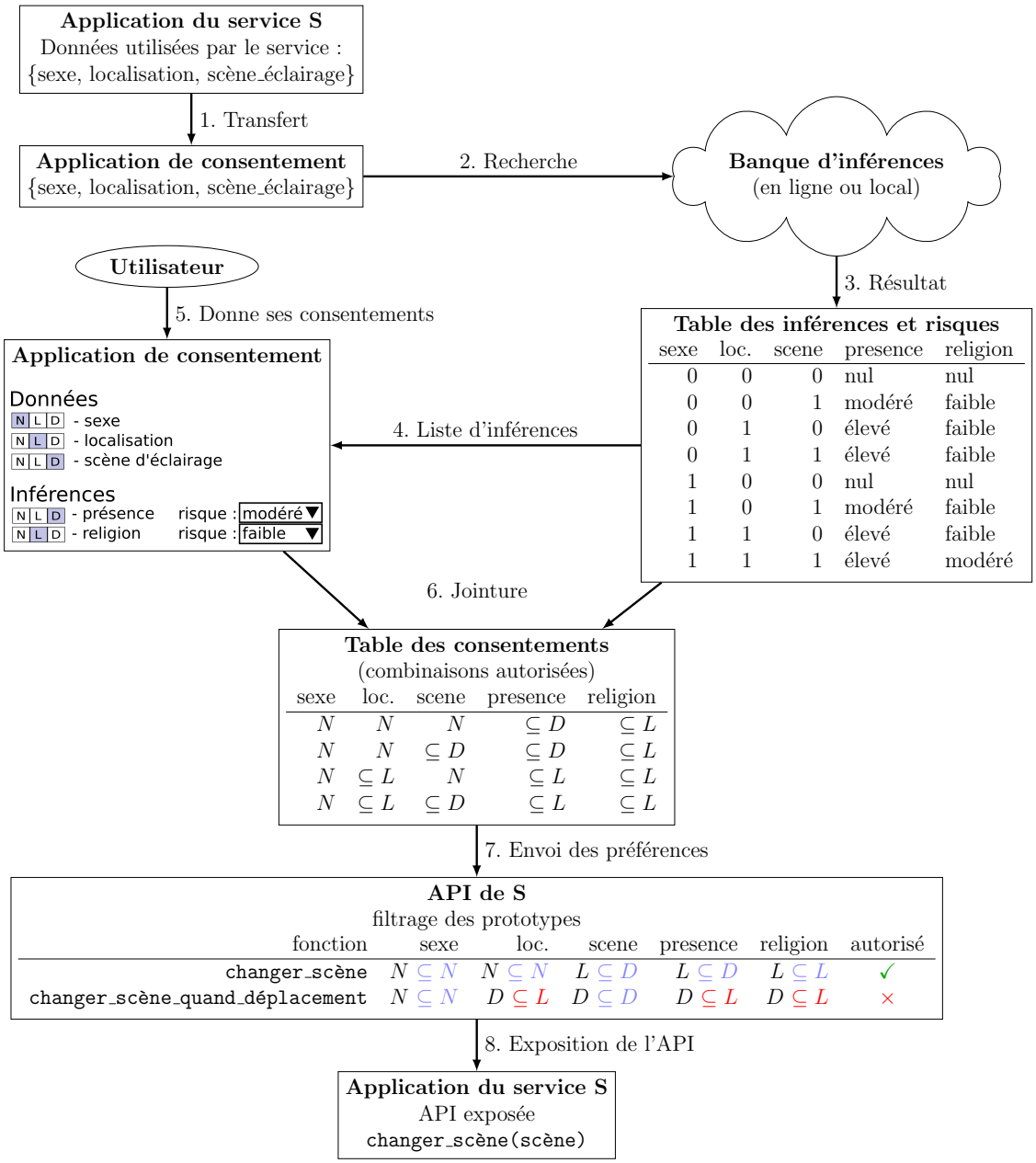


Figure 4.6: ***Fonctionnement des tables de consentements.*** *1. Une application redirige son utilisateur sur une application dédiée au consentement et transmet les types de données que S va capturer et/ou stocker. 2. L'application de consentement envoie ces types à une banque d'inférences. 3. La banque calcule toutes les combinaisons possibles de ces types avec les risques d'inférence associés. 4. Elle les envoie à l'application de consentement qui propose à l'utilisateur de donner ou non son consentement sur les données ([N]ul, [L]ocal, [D]istant) et sur les inférences en fonction de leur risque (nul, faible, modéré, élevé, certain). 5. L'utilisateur donne ses consentements. 6. L'application de consentement filtre les combinaisons compatibles avec les préférences de l'utilisateur. 7. Elle les envoie à S qui filtre les fonctions de son API selon si elles correspondent aux critères sélectionnés (en bleu si valide, rouge sinon) 8. S expose à l'utilisateur une API contenant uniquement les fonctions compatibles avec ses consentements.*

après leur fonctionnement. Voici tout d'abord la description des différents composants de la figure :

Données et inférences Je parle ici de données collectées et/ou stockées par une application, un objet ou un service associé. J'appelle inférence toute donnée qui n'est pas collectée mais qui peut être déduit des données collectées.

Risques d'inférences Ces risques représentent la probabilité qu'on puisse inférer la nouvelle donnée à partir de la combinaison de données fournie. La question du calcul de ce risque n'est pas traitée ici et mérite une étude à part entière. Les valeurs indiquées dans la figure ont été choisies uniquement à titre d'exemple.

Portée des autorisations L'utilisateur a le droit entre trois portées à autoriser : nul (0), local (L) et distant (D). Nul veut dire que l'utilisateur n'accepte aucune collècte de la donnée. Local indique qu'il accepte une collecte tant qu'elle n'est pas transmise sur Internet. Elle peut être conservée par exemple sur l'application ou dans un objet connecté. Distant signifique que la donnée peut être partagée sur Internet. On impose la propriété $0 \subseteq L \subseteq D$ sur les niveaux d'autorisation, ce qui permet de dire que lorsqu'un utilisateur consent au niveau D, alors il consent aussi aux niveaux L et 0. De la même manière, consentir au niveau L inclut le consentement au niveau 0, mais pas le niveau D.

Table des inférences et risques Table contenant les différentes combinaisons possibles d'un ensemble de données, ainsi que de toutes les inférences connues qu'on peut en déduire et les risques associés à ces dernières. Cette table est pré-calculée.

Table de consentement Table similaire à la table des inférences et risques, dont on a retiré toutes les combinaisons incompatibles avec les consentements de l'utilisateur, et dont les valeurs de présence dans la combinaison et de risques (0, 1, nul, faible, modéré, élevé, certain) sont remplacées par des intervalles (0, $\subseteq L$, $\subseteq D$) correspondant eux aussi aux volontés de l'utilisateur.

Application du service S Application utilisée par l'utilisateur qui sert à utiliser un service local ou distant. Exemple : Philips Hue est l'application du service Philips.

Application de consentement Application destinée à permettre à l'utilisateur de consentir ou non au partage de ses données et capable de générer une table de consentements. Les fonctions d'une API peuvent être filtrées grâce à cette table afin de ne garder que celles qui sont compatibles avec le consentement de l'utilisateur.

Banque d'inférences Service en ligne ou local qui fournit les inférences connues en fonction des données qu'on lui fournit. Son fonctionnement ne sera pas décrit ici et mérite une étude à part entière. Elle permet aussi de sauvegarder les préférences de l'utilisateur ou de les mettre à jour si besoin.

API Interface exposée par le service S contenant toutes les fonctions disponibles pour l'utilisateur. Les fonctions exposées seront celles qui correspondent à la table de consentement fournie par l'application de consentement.

La figure 4.6 propose différentes étapes pour permettre à un utilisateur de partager son consentement.

1. L'utilisateur lance une application dédiée à l'utilisation d'un service S. Si celle-ci est lancée pour la première fois, ou si les données collectées ou stockées par l'application, les objets et les services associés ont changé, alors l'application doit appeller une application dédiée au consentement et partager avec elle les types de ces données (dans la figure : `sexe`, `localisation` et `scène d'éclairage`).
2. L'application de consentement reçoit les types de données auxquelles l'utilisateur devra consentir. Elle envoie ces types de données à une banque d'inférence. Cette

banque d'inférence va chercher tous les types données qui peuvent être déduites à partir de n'importe quelle combinaison des types de données fournis par l'application de consentement.

3. La banque d'inférences génère une table des inférences dont elle a connaissance, ainsi qu'une estimation du risque d'obtention de ces inférences. Dans la figure, la banque a déterminé que les types de données `présence` et `religion` pouvaient être inférés. Par exemple, en connaissant la localisation de l'utilisateur, la banque d'inférences juge que le risque de pouvoir inférer que l'utilisateur est présent chez lui est `élevé`.
4. La table générée est envoyée à l'application de consentement, qui comprend alors que deux inférences sont possibles. Elle présente alors une interface avec la liste des données collectées, ainsi que celle des inférences possibles. Pour chaque donnée et inférence, l'application demande à l'utilisateur s'il accepte la collecte ou le stockage et à quel niveau (Nul, Local, Distant). Elle demande également le risque qu'il est prêt à prendre pour les inférences (nul, faible, modéré, élevé, certain).
5. L'utilisateur renseigne ses choix.
6. L'application de consentement génère une table des consentements en joignant les préférences de l'utilisateur à la table des inférences et risques. Cette procédure est expliquée en détail dans la figure 4.7.
7. La table des consentements est envoyée au service S afin que celui-ci filtre les fonctions de son API afin de ne conserver que celles qui sont compatibles avec les choix de l'utilisateur. Cette procédure est expliquée en détail dans la figure 4.8.
8. Le service S envoie à l'application (et aux objets associés) l'API à laquelle l'utilisateur a accès, ainsi on empêche le responsable de traitements de faire des actions en contradiction avec le consentement utilisateur.

Plusieurs améliorations sont possibles, la plus importante étant la sauvegarde des préférences de l'utilisateur, ce qui permet de réutiliser ses préférences pour d'autres services. Par exemple, on peut supposer que deux applications dédiées à l'éclairage d'une maison auront beaucoup de types de données en commun. Or les choix de l'utilisateur seront très probablement identiques pour ces deux applications, et sauvegarder ses préférences est une nécessité pour que l'utilisateur ait un avantage concret à utiliser l'application de consentement. Une autre amélioration liée à la précédente est le fait de spécifier un consentement « exceptionnel » pour un service particulier. Par exemple, si l'utilisateur partage sa localisation uniquement en local, il sera peut-être prêt à accepter de la partager avec un service distant s'il fait confiance à ce service. Il est aussi possible de donner, dans la table des inférences et risques, des coefficents aux données pour indiquer si une donnée a plus d'impact qu'une autre sur le risque d'inférence et ainsi augmenter le niveau de détail des tables de consentement. Dernière amélioration importante : ajouter un niveau d'autorisation entre `local` et `distant` qui serait dédié aux échanges locaux mais avec des services différents que l'utilisateur pourrait spécifier. Par exemple, il pourrait utiliser openHAB pour gérer son éclairage, mais refuser d'utiliser d'autres service comme Google Assistant.

Grâce à cette proposition, on est donc en capacité de générer une « table de consentements » qui pourrait être partagée parmi les multiples services utilisés par un individu, sans qu'il ait à reformuler ses consentements à chaque nouveau service. De plus cela peut permettre d'avoir une base de connaissance partagée par les fabricants, les utilisateurs et les chercheurs, afin qu'ils soient tous aussi bien informés sur la pertinence de la collecte d'un certain type de donnée et des risques associés. Cette table pourrait être mise à jour à chaque fois qu'une nouvelle inférence est trouvée par la recherche.

Comme on le voit, la prise en charge efficace du consentement contraint obligatoirement les fabricants à repenser intégralement le développement de leurs services. L'exposition de

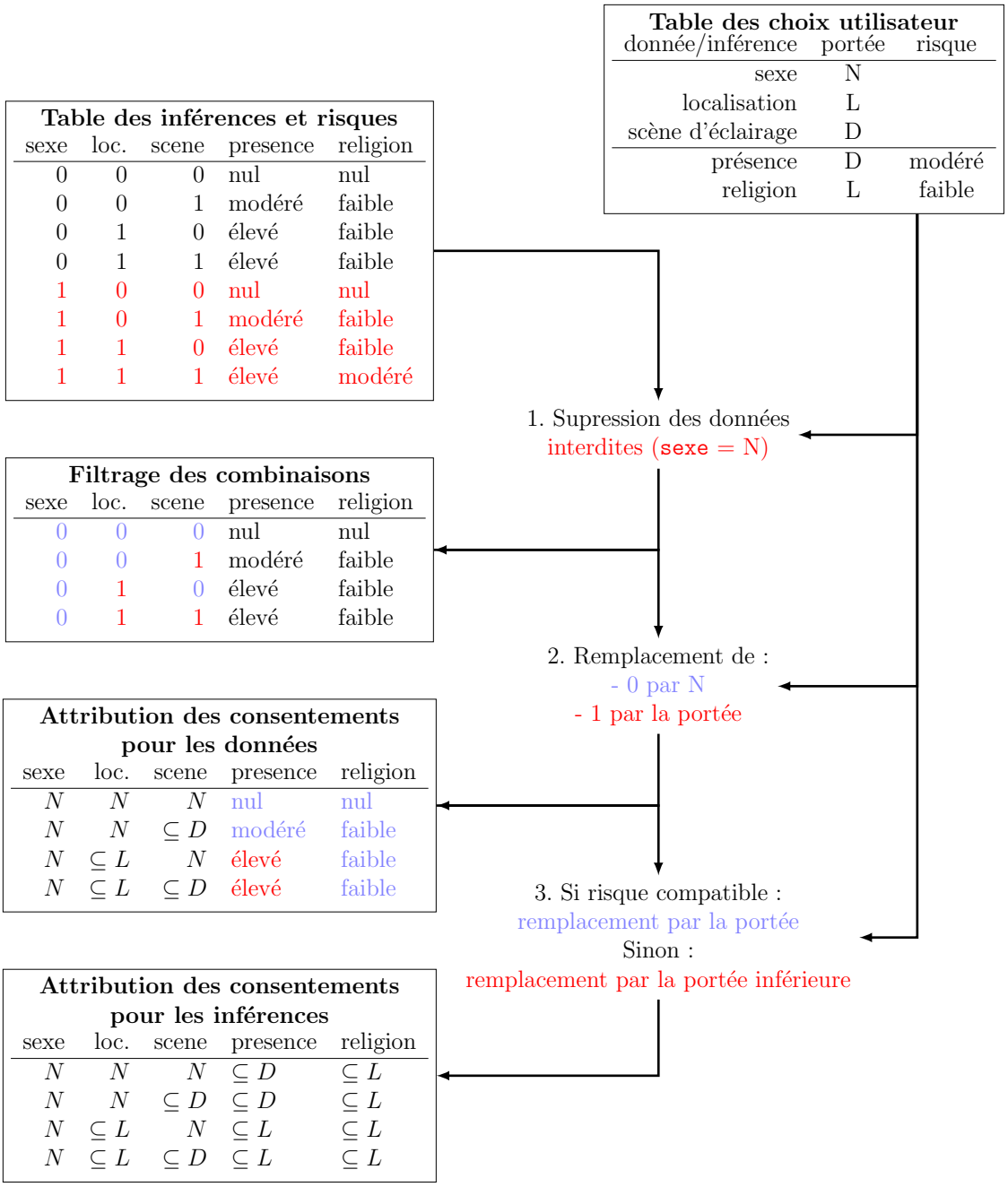


Figure 4.7: ***Génération d'une table des consentements.*** *Dans l'étape 3, comme le risque d'obtenir l'inférence « présence » est élevé dans le cas 3 et 4, et que la portée définie par l'utilisateur est D, on se rabat sur la portée inférieure : L.*

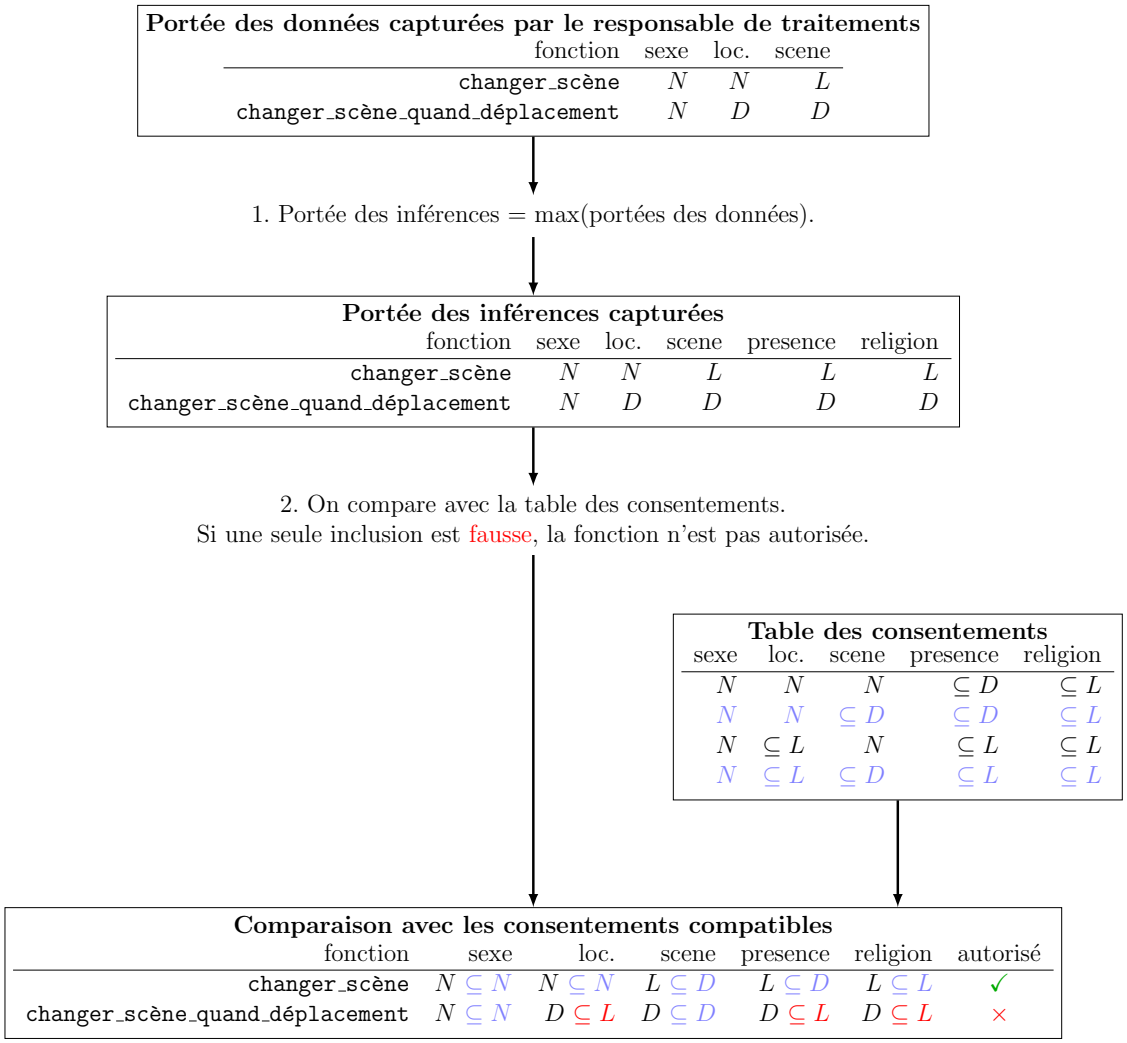


Figure 4.8: ***Filtrage des prototypes d'une API*** *Les portées des données capturées sont déterminées par le responsable de traitements. Les seules combinaisons compatibles avec les données utilisées par les fonctions sont indiquées en bleu.*

l'API du fabricant dépend du consentement de l'utilisateur et les développeurs devront savoir pour chaque fonction, quels types de données seront utilisés.

Chapitre 5

Conclusion et perspectives

L'objectif de ce chapitre est de donner ma vison générale sur l'état de la recherche en matière de vie privée et de sécurité concernant l'Internet des objets.

Je discuterai des différents problèmes que j'ai pu rencontrer lors de mes travaux, et ferai plusieurs remarques et critiques qui selon moi représentent les principaux axes de recherche à aborder aujourd'hui.

J'ai divisé ces réflexions en plusieurs parties. En section 5.1, je parlerai de la grande complexité de l'Internet des objets et de ses implications sur la responsabilité des fabricants et des utilisateurs en matière de vie privée et de sécurité. En section 5.2, je discuterai sur le paradoxe de la sécurité déployée par les fabricants, qui protège l'utilisateur contre les attaquants, mais ralentie l'avancée de la recherche en sécurité. En section 5.3, j'expliquerai pourquoi, selon moi, la distinction juridique entre données à caractère personnel et données sensibles n'est plus suffisante aujourd'hui. En section 5.4, je parlerai de la nécessité de développer des outils communs entre les fabricants, les utilisateurs et les chercheurs, afin de faciliter la compréhension, la collecte, et la vérifiabilité du consentement. En section 5.5, je discuterai du manque de systèmes de détection des utilisateurs multiples et des « non-utilisateurs » (par exemple un voisin) dont les données sont capturées malgré eux. Enfin, je terminerai en section 5.6 sur la question de l'écologie et de l'énergie consommée par l'Internet des objets.

5.1 Vie privée : une affaire complexe, quelles solutions ?

La complexité de l'Internet des objets est impressionnante. Les interactions entre les objets, leurs capacités, les risques qui leurs sont associés sont multiples. On constate d'abord une hétérogénéité des protocoles, dont les buts sont parfois radicalement différents : économie d'énergie, sécurité, accessibilité. Un utilisateur ne peut pas raisonnablement être en capacité de comprendre les mécanismes sous-jacent à ces protocoles. D'autre part, du point de vue de la vie privée, même les protocoles les plus sécurisés ne sont pas toujours mis en place de la bonne façon par les fabricants. Par exemple, il n'est pas forcément connu de tous qu'un protocole, même sécurisé au moyen d'un chiffrement par exemple, peut laisser transparaître un comportement utilisateur. Il faut donc être à la pointe de la recherche en permanence pour connaître toutes les techniques d'inférence d'information basées sur les réseaux, ce qui n'est le cas ni des utilisateurs, ni des fabricants.

Les protocoles sont au coeur de l'Internet des objets, mais une grande partie des problèmes qu'on peut y rencontrer est liée aux objets connectés, aux applications et aux services fournis par les fabricants de ces derniers. Or on constate encore une fois que la vie privée n'est pas un domaine simple à appliquer, en partie à cause de son évolution

constante. Comme nous avons pu le voir, que ce soit sur le plan technique (protocoles, sécurité, nature et souveraineté des données transmises, inférences, traqueurs) ou sur le plan juridique (RGPD, consentement, chartes de vie privée), les erreurs sont faciles à commettre. Et là encore, tout cette complexité est inaccessible à l'utilisateur, qui n'a de toute façon pas à s'y intéresser. En revanche, il est de la responsabilité des fabricants d'éviter ce genre de problème afin que les utilisateurs puissent leur faire confiance.

Donner des responsabilités à l'utilisateur devient alors discutable. Mais alors à qui incombent-elles ? L'utilisateur doit conserver un pouvoir de consentement sur ce qui est fait de ses données, mais pour cela, il faut qu'il soit bien informé. L'information qu'on lui fournit doit être suffisamment claire, explicite et simple, afin que son consentement soit considéré comme valide, ce qui n'est pas une tâche aisée. Il revient en partie au fabricant de fournir ce type d'information. De plus, l'aspect technique ne peut évidemment pas être géré par l'utilisateur, donc là aussi, on se tourne vers le fabricant. Mais tout comme les problèmes cités précédemment, il peut être difficile, même pour un fabricant, de bien faire les choses. J'ai présenté ici un bon nombre d'erreurs commises par les fabricants, dont même les plus grands, qui ont le plus de moyens financiers (Google, Amazon) ne sont pas exempts. Dès lors, si même ceux qui ont le plus de moyens ne peuvent fournir une solution suffisante pour garantir la vie privée de l'utilisateur et recueillir un consentement valide, je ne pense pas qu'il soit raisonnable de penser que les petits fabricants en soient capables.

Pourtant, du fait de l'introduction du RGPD et de l'avancée de la recherche, concevoir des produits et services respectueux de la vie privée devient de plus en plus pressant et nécessite d'engager des experts du domaine. Or, même si cela est possible pour les grandes entreprises, ce n'est certainement pas le cas des petites. Avoir un expert de la vie privée pourrait bien devenir la norme dans les entreprises qui traitent des informations à caractère personnel. Depuis la mise en place de RGPD, on a vu apparaitre un nouveau métier, celui de DPO (délégue à la protection des données) chez les responsables de traitements. En France, la CNIL peut offrir son aide ponctuellement sur certaines questions, mais avec les besoins croissants des entreprises en matière de vie privée, je doute que cette commission puisse suivre la cadence. Si on considère la vie privée comme un droit fondamental, il serait donc bénéfique de trouver des solutions pour permettre à tous les fabricants de respecter au mieux la vie privée de leurs utilisateurs. Il pourrait aussi être utile de créer un système de certification pour garantir à l'utilisateur un certain niveau de sécurité et de respect de la vie privée, ce qui peut aussi inciter les fabricants à mettre en oeuvre les moyens suffisant pour obtenir cette certification. Comment certifier à un utilisateur qu'un fabricant respecte sa vie privée au mieux reste un sujet à explorer.

Lors de nos travaux, un problème s'est révélé très fréquent et mérite selon moi une attention particulière : l'inférence d'activité à partir du trafic réseau. Il est relativement simple actuellement de déduire ce que fait un utilisateur en se basant sur le trafic réseau qu'il génère, même si celui-ci est chiffré. Pour cette raison, il est nécessaire de créer des techniques de dissimulation d'activité réseau. Ce problème est extrêmement difficile à corriger. Plus la dissimulation fonctionne, plus la bande passante et les ressources matérielles nécessaires augmentent, ce qui impose de nouvelles limites physiques. Ces limites doivent d'ailleurs être revues à la baisse si on souhaite économiser l'énergie.

Si on cherche à empêcher un attaquant de détecter l'activité de l'utilisateur, rechercher le meilleur compromis entre dissimulation d'activité et consommation de bande passante (cas où de la génération de faux traffic pour brouiller les pistes) et ressources est une priorité. Cela implique de donner des solutions aux fabricants pour dissimuler leurs transmissions de données, et créer des outils de détection d'activité pour prouver que les solutions des fabricants fonctionnent.

Axes de recherche :

1. Aider les fabricants, peu importe leur taille, à recueillir un consentement valide de

l'utilisateur.

2. Étudier comment informer au mieux l'utilisateur de ce que les données qu'il partage peuvent réveler sur lui.
3. Permettre aux fabricants de protéger leurs services contre l'inférence d'activité basée sur le trafic réseau.

5.2 Paradoxe de la recherche en vie privée

Lors de mes travaux, j'ai pu remarquer un paradoxe concernant la vie privée et la sécurité dans le monde de la recherche. Les avancées en la matière nous permettent de toujours améliorer la situation en réparant les problèmes découverts au fur et à mesure. Ce faisant, on rend la tâche toujours plus difficile à de potentielles entités malicieuses désireuses d'attaquer des systèmes informatiques ou des données. Le paradoxe vient du fait que, pour qu'un chercheur puisse étudier, critiquer ou améliorer un système, il doit utiliser des techniques similaires à ces attaquants, à la différence qu'il doit s'engager à ne pas nuire au système. Or toutes les protections mises en oeuvre pour lutter contre les attaques deviennent en même temps des obstacles inutiles pour un chercheur bien intentionné qui perdra un temps précieux en essayant de les contourner.

Les fabricants font également leur part de recherche, ou utilisent les dernières avancées du monde académique pour améliorer leur sécurité et leur respect de la vie privée. Ils peuvent alors créer des services efficaces de ce point de vue, mais faire une confiance aveugle à leurs capacités n'est pas possible. Une grande partie des travaux de cette thèse consiste d'ailleurs à montrer que, honnêteté à part, le résultat final n'est pas à la hauteur de nos attentes. Mais pour en arriver à ce résultat, nous avons dû nous confronter aux mécanismes de défense déployés par les fabricants, comme de véritables attaquants le feraient. Même en considérant les meilleurs experts, si le fabricant ajoute une couche de sécurité de sa propre conception, ces obstacles peuvent toujours survenir, et cela ralenti fortement la recherche. Indirectement, en implémentant des mécanismes pour respecter la vie privée de leur utilisateurs, les fabricants ralentissent la recherche sur comment mieux la respecter.

Pour donner un exemple, il est assez courant pour les fabricants de faire usage d'obfusaction de données, par chiffrement par exemple. Le problème ici est que, bien que les données soient protégées des attaques, un chercheur ne peut vérifier si l'obtention par le fabricant des données obfusquées est pertinent ou légitime.

L'usage d'ingénierie inversée est aussi assez controversée puisqu'elle peut permettre de découvrir des mécanismes déployées par les fabricants qui ne souhaitent pas partager leur technologie. Les chercheurs et les attaquants peuvent tous appliquer ces techniques mais encore une fois, ce sont les chercheurs qui en pâtissent.

Toujours lors de nos travaux, nous nous sommes confrontés à un autre problème, cette fois-ci lié à la dynamicité des services sur Internet, et non à la sécurité. Nous avons eu l'occasion d'essayer de vérifier la destination des données, ce qui s'est avéré beaucoup plus complexe qu'escompté. Nous avons pu essayer différents outils pour déterminer ces destinations, dont certains conseillés par des experts. Aucun ne s'est révélé être suffisamment efficace, chacun d'entre eux a une couverture et des résultats différents, et nous n'avons trouvé aucun moyen de confirmer quel résultat était correct. Or c'est un besoin essentiel pour confirmer que la souveraineté du service est conservée. On retrouve à nouveau ce paradoxe de la recherche contre les fabricants. Ces derniers peuvent utiliser des techniques pour qu'on ne puisse pas simplement tracer les données, mais cette difficulté est aussi présente pour les chercheurs.

Pour régler ce paradoxe, il pourrait y avoir plusieurs solutions. La première, assez naïve,

serait de permettre aux chercheurs d'avoir des accès particuliers pour qu'ils puissent travailler facilement. Cela n'est pas envisageable et rappelle les tentatives de la police d'avoir des accès dérobés dans certains service pour avoir le pouvoir de détecter des utilisateurs malintentionnés. Or il est assez connu que fournir de tels accès ne permet pas de garantir que seuls les bénéficiaires prévus par le fabricant pourront en bénéficier, cela ajoute alors un nouveau problème de sécurité dans le cas où des attaquants mettraient la main sur ces accès. De plus, cela ne peut pas permettre de garantir à coup sûr que le comportement observé en utilisant les accès fournis soient représentatif du comportement qu'on observerait en tant qu'utilisateur.

Une deuxième solution serait de concevoir des outils pour tracer les données générées par les applications et les objets connectés. Lors de mes travaux, j'ai eu l'occasion d'implémenter l'ébauche d'un système de traçage des données sensibles sur smartphone grâce à Frida. Le but était d'être capable de suivre une donnée du moment où elle est générée par le système d'exploitation, jusqu'au moment où elle quitte l'application. Avec Frida, il est possible d'intercepter certains appels systèmes ou des appels à l'API Android et ainsi détecter quand, par exemple, une géolocalisation est récupérée, si celle-ci passe par l'API de chiffrement d'Android, en récupérer la version chiffrée, et voir si elle est envoyée hors de l'application et à quelle adresse.

Un tel outil pourrait grandement aider à analyser les applications d'un point de vue vie privée, et de manière automatique. Frida est un projet encore assez récent, mais offre déjà des possibilités suffisantes pour accomplir de beaux projets. Malheureusement, je n'ai pas eu le temps d'achever ce projet, mais je pense que ce genre d'outil devient essentiel de nos jours, pour redonner un peu de transparence à ce que font les applications.

À l'heure des objets connectés, on se retrouve confronté à un problème similaire : la transparence dans les algorithmes utilisés par les objets connectés. Un outil similaire à celui que j'ai commencé de développer, mais cette fois appliqué aux les objets connectés, serait aussi très utile. Comme il nécessiterait de s'attaquer à la partie matérielle (l'objet connecté), Frida ne serait d'aucune utilité ici. C'est un domaine à part entière que je n'ai pas abordé pendant ma thèse, mais qui aurait un impact significatif sur la transparence.

Cela m'amène donc à un problème plus large : celui de la transparence des fabricants et de leur imputabilité. En effet, je ne vois pas de moyens propre et honnête d'obtenir les informations nécessaires sur un système concernant sa façon de protéger la vie privée, sans que les fabricants y consentent pleinement, et cela ne peut se faire qu'en étant totalement transparents avec le public (utilisateurs et chercheurs) sur comment leur service fonctionne. Il faut aussi qu'on puisse punir les fabricants qui prétendraient faire quelque chose, et en feraient une autre. Mais pour qu'un fabricant puisse être totalement transparent, il faut qu'il ait un intérêt à le faire. Là encore, un système de certification pourrait être bénéfique. Il reste selon moi beaucoup de recherche à effectuer sur ces problèmes de paradoxe entre la volonté des fabricants de protéger les intérêts de l'utilisateur et les siens en même temps.

Axes de recherche :

1. Création d'outils pour tracer les données générées par des applications ou objets connectés, de leur création à leur transmission.
2. Inciter les fabricants à faire preuve de plus de transparence concernant leur fonctionnement de leurs services.

5.3 Données simples, à caractère personnel et sensibles

La distinction entre données simples, données à caractère personnel et données sensibles représente un autre problème selon moi. Les fabricants se servent de ces notions pour

justifier ou non un traitement de données, pour anonymiser ces données si besoin, ou pour les transmettre à l'utilisateur. Si je passe par Google Takeout pour récupérer les données que Google a sur moi par exemple, il est assez clair pour moi que se limiter aux données à caractère personnelles n'est pas suffisant.

Le premier problème que je vois dans ce cas de figure est que je considère que les données simples n'existent tout simplement pas. En théorie, une donnée qui n'est pas à caractère personnel est une donnée qui n'est pas relative à une personne physique susceptible d'être identifiée, directement ou indirectement. En théorie toujours, cela pourrait être par exemple un âge ou un sexe par exemple. Or, ce type de données peut permettre, suite à une agrégation d'autres données, d'identifier une personne. D'ailleurs plus la recherche avance dans ce domaine, plus on se rend compte que quasiment la totalité des données, si anodines qu'elles soient, peuvent donner un indice sur l'identité de la personne.

Le deuxième problème est que le caractère sensible d'une donnée est basé sur des problèmes de société liés à une époque particulière. Cela implique que les traitements valables aujourd'hui ne le seront peut-être plus demain, et que pour chaque changement, les fabricants doivent se mettre à jour. Dans le cas de changements d'échelle massif en terme d'application des loi, comme on l'a vu avec le RGPD, cela peut entraîner des ruptures dans le fonctionnement de certains services, ce qui n'est pas souhaitable. À la rigueur, on peut ignorer ce problème car de tels changements se produisent assez rarement. Mais selon moi, il y a un second problème qui mérite d'être résolu : un utilisateur peut aussi considérer une donnée comme délicate pour lui, bien qu'elle ne le soit pas pour d'autres. Ces données ne sont donc pas forcément vues comme sensibles aux yeux de la loi, alors qu'elles posent un réel problème à leur propriétaire.

Faire de la recherche premièrement sur les inférences possibles sur les données, deuxièmement sur les questions éthiques liées aux données, me parait primordial. Je ne connais pas de solution qui puisse satisfaire tout le monde sur ces deux problématiques, si ce n'est permettre aux utilisateurs de personnaliser le traitement qui est appliqué à leurs données, et ce avec un niveau de détail raisonnable. Pour cela, je pense qu'il faut commencer par considérer les données non pas en trois catégories (simples, à caractère personnel et sensible), mais comme un curseur qu'on peut placer sur trois axes : risque d'identification, sensibilité juridique et « délicatesse » ou « ressenti » personnel. Les deux premiers axes pourraient être déterminés par des chercheurs en inférences de données et par des juristes. Le troisième axe serait par défaut équivalent au second, mais serait personnalisable par l'utilisateur. Un outil commun similaire à celui proposé en fin du chapitre 4 sera alors obligatoire.

Axes de recherche :

1. Développer la notion de donnée délicate et ses implications d'un point de vue juridique et personnel.
2. Étudier comment permettre aux fabricants de donner à l'utilisateur la totalité de ses données personnelles (y compris délicates) de manière sécurisée.

5.4 Consentement et outils communs

On ne peut pas espérer qu'un utilisateur soit suffisamment éduqué sur la question de la vie privée, mais il est important qu'il le soit un minimum et qu'il ait les capacités de la conserver tant que possible. J'ai présenté un système de tables de consentement en chapitre 4 qui pourrait être une première étape vers un outil commun aux fabricants et aux utilisateurs. Le problèmes des inférences de données cité précédemment concerne aussi ces outils, puisqu'il faut que fabricants et utilisateurs soient capables de comprendre quelles sont les risques associés aux données qu'ils capturent ou partagent, ainsi que la sensibilité

que ces données ont.

Unifier cela au moyen de structures de données standardisées me semble être un des axes de recherche les plus urgents actuellement. Repenser le développement des applications et objets afin qu'ils prennent en compte naturellement les choix de l'utilisateur en matière de consentement en fait partie. Le système actuel me paraît trop rigide et trop peu adapté aux besoins actuels et futurs. Les API sont, me semble-t-il, trop axées sur les fonctionnalités, sans se préoccuper suffisamment de l'éthique et des valeurs de l'utilisateur. Il peut être difficile d'avoir une compréhension de ce qu'est le consentement, preuve en est les interfaces trompeuses que j'ai présentées dans cette thèse. Avoir un outil commun entre fabricants, chercheurs et utilisateurs réduirait également ce genre de problème.

Une structure de données standardisée partagée entre les utilisateurs et les fabricants serait selon moi une avancée considérable du point de vue du respect des volontés de l'utilisateur. Une telle structure permettrai aussi de faciliter la vérification de la conformité au RGPD des fabricants. Il serait aussi possible d'accompagner cette structure d'algorithmes permettant de déduire automatiquement les données auxquelles l'utilisateur consent. De cette manière, il serait possible de faciliter grandement le consentement de l'utilisateur qui n'aurait plus à choisir chaque donnée qu'il souhaite partager indépendamment, réduisant ainsi l'effort à fournir.

Axes de recherche :

1. Créer une structure de données commune aux fabricants et utilisateurs pour le partage de consentement.
2. Créer des outils de détection automatique d'inférences.
3. Permettre aux utilisateurs de consentir ou non à des données obtenues par inférence.

5.5 Utilisateurs multiples et passant

La gestion des utilisateurs multiples soulève des questions d'ordre technique et juridique qui sont à ma connaissance peu traitées aujourd'hui. Gérer le consentement des différents membres d'une famille n'est actuellement pas possible sur les applications et objets que j'ai pu tester, et les implications en matière de fonctionnalités sont vastes.

La gestion des utilisateurs passants est aussi un axe de recherche intéressant. Encore une fois, avoir un outil commun de gestion du consentement pourrait permettre à n'importe qui de facilement transmettre sa « table de consentement » rapidement à un objet ou une application. Ces derniers auront par contre la responsabilité de vérifier qui est en train de les utiliser, et de permettre à chacun de spécifier ce à quoi il consent, facilement. On peut imaginer des phrases sur un certains modèle pour les enceintes connectées, des gestes pour des appareils avec webcam, ou encore une combinaison de touches par exemple.

Axes de recherche :

1. Étudier le consentement dans le cas d'utilisateurs multiples.
2. Améliorer la distinction entre les utilisateurs (et non-utilisateurs) d'un objet.

5.6 Internet des objets et écologie

L'impact écologique de la maison intelligente est un autre sujet qui reste à traiter dans le cadre de travaux futurs. Nous avons vu que les techniques de contrôle basées sur un assistant vocal génèrent un ou deux ordres de grandeur de données supplémentaires, envoyées sur Internet, que les autres techniques. Indépendamment des enceintes intelligentes, le simple fait de s'appuyer sur les communications Internet génère des coûts énergétiques supplémentaires (transmission, routage et stockage) qui doivent être ajoutés à ceux liés à

l'alimentation continue des appareils. Il est important d'estimer les impacts réels de ces effets secondaires, en utilisant une approche scientifique.

Bien qu'un des buts premiers de certains objets connectés soit la diminution du coût énergétique, certains articles essayent de déterminer les consommations réelles de ces objets, les transmissions de données, leur stockage, les différences entre traitement local et distant des données [189, 190]. Force est de constater que la situation n'est pas simple du tout. La consommation d'un seul appareil chez l'habitant est-elle comparable à la consommation d'un appareil situé dans un autre pays mais capable d'adapter sa consommation en fonction des besoins des clients ? Rien n'est sûr.

Ce qui est sûr en revanche, c'est qu'il est nécessaire de disposer d'architectures et de technologies qui favorisent le traitement et les communications locales si nous voulons que la maison intelligente soit durable et acceptable, et que la souveraineté des données soit respectée. Si les appareils fonctionnant localement s'avéraient suffisamment efficaces d'un point de vue énergétique, il pourrait alors y avoir une convergence entre la vie privée, la souveraineté et l'écologie.

Axes de recherche :

1. Étudier l'impact des objets connectés sur la consommation électrique.
2. Étudier la différence d'impact entre des traitements locaux, en bordure ou dans le cloud.

Bibliographie

[1] B. Schneier, "Crypto-Gram: Security and the Internet of Things," févr. 2017.

[2] "Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016."

[3] L. Columbus, "10 Charts That Will Challenge Your Perspective Of IoT's Growth."

[4] J. Titcomb, "Mobile web usage overtakes desktop for first time," *The Telegraph*, nov. 2016.

[5] "Gartner Says by 2019, 20 Percent of User Interactions With Smartphones Will Take Place via VPAs."

[6] P. P. A. p. W. I. |. M. . Mai 2018, "Google Home passe devant Amazon Echo."

[7] "Google Wants To Data Mine Your Home And Kids' Bedroom | Zero Hedge."

[8] Bastien, "Amazon Echo peut envoyer vos conversations à vos contacts à votre insu," mai 2018.

[9] "Apple aussi écoute vos conversations avec Siri (sans votre consentement)."

[10] S. Maheshwari, "Burger King "O.K. Google" Ad Doesn't Seem O.K. With Google," *The New York Times*, avr. 2017.

[11] A. Liptak, "Amazon's Alexa started ordering people dollhouses after hearing its name on TV," janv. 2017.

[12] P. Oltermann, "German parents told to destroy doll that can spy on children," *The Guardian*, févr. 2017.

[13] R. Jones, "Roomba's Next Big Step Is Selling Maps of Your Home to the Highest Bidder."

[14] N. Vallina-Rodriguez et S. Sundaresan, "7 in 10 smartphone apps share your data with third-party services."

[15] "Amazon will stop selling connected toy filled with security issues."

[16] G. R, "Google Home et Chromecast victimes d'une faille de sécurité," juin 2018.

[17] A. Greenberg, "Hackers Remotely Kill a Jeep on the Highway–With Me in It," juill. 2015.

[18] "Mirai (malware)," mai 2017.

[19] S. Chaptal, "Shodan : un moteur de recherche rêvé pour cybercriminels, et les responsables IoT," sept. 2016.

[20] "2018 reform of EU data protection rules."

[21] P. LABBE, "RGPD : de nombreux objets connectés en panne avec la nouvelle législation," mai 2018.

[22] D. Basulto, "This amazing remote-controlled contraceptive microchip you implant under your skin is the future of medicine," juill. 2014.

[23] "Portal from Facebook: Voice Enabled Hands-Free Video Calling."

[24] D. Schweppe, "Mycroft – Open Source Voice Assistant."

[25] R. Hindi, ""Hey Snips!" – Announcing the first Private-by-Design Voice Platform," juin 2017.

[26] "Sonos Announces Acquisition of Snips."

[27] H. Derhamy, J. Eliasson, J. Delsing et P. Priller, "A survey of commercial frameworks for the Internet of Things," dans *IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, sept. 2015, p. 1–8.

[28] S. Marksteiner, V. J. E. Jimenez, H. Valiant et H. Zeiner, "An overview of wireless IoT protocol security in the smart home domain," dans *Internet of Things Business Models, Users, and Networks*, nov. 2017, p. 1–8.

[29] M. Chan, D. Esteve, J.-Y. Fourniols, C. Escriba et E. Campo, "Smart wearable systems: Current status and future challenges," *Artificial Intelligence in Medicine*, vol. 56, n°. 3, p. 137–156, nov. 2012.

[30] H. N. Saha, A. Mandal et A. Sinha, "Recent trends in the Internet of Things," dans *IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, janv. 2017, p. 1–4.

[31] "Wi-Fi Direct," mars 2020.

[32] "Thread (network protocol)," avr. 2017.

[33] "Radio-frequency identification," mars 2020.

[34] H. Haas et C. Chen, "What is Li-Fi ?" 2015, p. 1–3.

[35] "ZigBee," avr. 2017.

[36] "Z-Wave," mars 2017.

[37] "OSI model," mars 2020.

[38] "Wi-Fi," mars 2020.

[39] "Ethernet," mars 2020.

[40] "3G," mars 2020.

[41] "4G," mars 2020.

[42] "5G," avr. 2020.

[43] A. Fendelman, "1G, 2G, 3G, 4G, & 5G Explained," 2019.

[44] "IEEE 802.15.4," mai 2016.

[45] "Medium access control," mars 2020. [En ligne]. Disponible : https://en.wikipedia.org/w/index.php?title=Medium_access_control&oldid=947021239

[46] J. Postel, "Internet Protocol," 1981.

[47] R. Hinden et S. Deering, "Internet Protocol, Version 6 (IPv6) Specification," 1995.

[48] N. Kushalnagar, G. Montenegro, D. E. Culler et J. W. Hui, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," 2007.

[49] Y. Dalal, C. Sunshine et V. Cerf, "Specification of Internet Transmission Control Program," 1974.

[50] J. Postel, "User Datagram Protocol," 1980.

[51] T. Dierks et C. Allen, "The TLS Protocol Version 1.0," 1999.

[52] N. Modadugu et E. Rescorla, "Datagram Transport Layer Security," 2006.

[53] H. F. Nielsen, T. Berners-Lee et R. T. Fielding, "Hypertext Transfer Protocol – HTTP/1.0," 1996.

[54] E. Rescorla, "HTTP Over TLS," 2000.

[55] "MQTT," mars 2020.

[56] Z. Shelby, K. Hartke et C. Bormann, "The Constrained Application Protocol (CoAP)," 2014.

[57] "Bluetooth," mars 2020.

[58] "Bluetooth Low Energy," févr. 2020.

[59] "Our History."

[60] "Near-field communication," mars 2020.

[61] N. Sornin, M. Luis, T. Eirich, T. Kramp et O. Hersent, "LoRaWAN Specification," 2015.

[62] "Sigfox Device Radio Specifications | Sigfox build."

[63] "EnOcean," sept. 2019.

[64] "KNX (standard)," févr. 2020.

[65] "A History of KNX."

[66] "About LoRaWAN® | LoRa Alliance®."

[67] "IEEE 802.3," mars 2020.

[68] "IEEE 802.11," avr. 2017.

[69] "IEEE 802.15," janv. 2020.

[70] "ISO/IEC 14443," févr. 2020.

[71] 14 :00-17 :00, "ISO/IEC 14543-3-10:2012."

[72] S. Imtiaz, R. Sadre et V. Vlassov, "On the Case of Privacy in the IoT Ecosystem: A Survey," dans *International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, juill. 2019, p. 1015–1024.

[73] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier et P. Kikiras, *On the Security and Privacy of Internet of Things Architectures and Systems*, sept. 2015.

[74] D. Leibenger, F. Möllers, A. Petrlic, R. Petrlic et C. Sorge, "Privacy Challenges in the Quantified Self Movement – An EU Perspective," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, n°. 4, p. 315–334, 2016.

[75] "Google Takeout."

[76] "Frida A world-class dynamic instrumentation framework." [En ligne]. Disponible : https://www.frida.re/

[77] A. Rao, A. M. Kakhki, A. Razaghpanah, A. Tang, S. Wang, J. Sherry, P. Gill, A. Krishnamurthy, A. Legout et A. Mislove, "Using the middle to meddle with mobile," *CCIS, Northeastern University, Tech. Rep., December*, 2013.

[78] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman et V. Paxson, "Haystack: A Multi-Purpose Mobile Vantage Point in User Space," *arXiv preprint arXiv :1510.01419*, 2015.

[79] A. Le, J. Varmarken, S. Langhoff, A. Shuba, M. Gjoka et A. Markopoulou, "AntMonitor: A System for Monitoring from Mobile Devices," dans *Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data - C2B(1)D '15*. ACM Press, 2015, p. 15–20.

[80] guenael@nextinpact.com, "Avec PiRogue, capturer le trafic HTTP(S) d'un smartphone devient plus simple," mars 2018.

[81] "mitmproxy - an interactive HTTPS proxy."

[82] "Burp Suite Scanner | PortSwigger."

[83] "Princeton IoT Inspector."

[84] S. Siby, R. R. Maiti et N. Tippenhauer, "IoTScanner: Detecting and Classifying Privacy Threats in IoT Neighborhoods," *arXiv preprint arXiv :1701.05007*, 2017.

[85] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. R. Sadeghi et S. Tarkoma, "IoT SENTINEL : Automated Device-Type Identification for Security Enforcement in IoT," dans *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, juin 2017, p. 2177–2184.

[86] "Fast Web-based Attacks to Discover and Control IoT Devices."

[87] D. Wood, N. Apthorpe et N. Feamster, "Cleartext Data Transmissions in Consumer IoT Medical Devices," dans *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*. ACM, 2017, p. 7–12.

[88] J. Ren, A. Rao, M. Lindorfer, A. Legout et D. Choffnes, "ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic." ACM, juin 2016, p. 361 – 374.

[89] T. OConnor, R. Mohamed, M. Miettinen, W. Enck, B. Reaves et A.-R. Sadeghi, "HomeSnitch: Behavior Transparency and Control for Smart Home IoT Devices," dans *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '19. ACM, 2019, p. 128–138.

[90] N. Apthorpe, D. Reisman et N. Feamster, "A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic," *arXiv :1705.06805 [cs]*, mai 2017.

[91] B. Copos, K. Levitt, M. Bishop et J. Rowe, "Is Anybody Home ? Inferring Activity From Smart Home Network Traffic," dans *IEEE Security and Privacy Workshops (SPW)*, mai 2016, p. 245–251.

[92] P.-M. Junges, J. François et O. Festor, "Passive Inference of User Actions through IoT Gateway Encrypted Traffic Analysis," dans *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, p. 7–12.

[93] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi et A. S. Uluagac, "Peek-a-Boo: I see your smart home activities, even encrypted!" *arXiv :1808.02741 [cs]*, août 2018.

[94] L. Caviglione, A. Merlo et M. Migliardi, "Covert Channels in IoT Deployments Through Data Hiding Techniques," mai 2018, p. 559–563.

[95] A. Maiti et M. Jadliwala, "Light Ears: Information Leakage via Smart Lights," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, n°. 3, p. 98 :1–98 :27, sept. 2019.

[96] E. Ronen et A. Shamir, "Extended Functionality Attacks on IoT Devices: The Case of Smart Lights," dans *IEEE European Symposium on Security and Privacy (EuroS&P)(EUROSP)*, mars 2016, p. 3–12.

[97] L. Olejnik et A. Janc, "Stealing sensitive browser data with the W3C Ambient Light Sensor API," avr. 2017.

[98] H. Wang, T. T.-T. Lai et R. Roy Choudhury, "MoLe: Motion Leaks Through Smartwatch Sensors," dans *21st Annual International Conference on Mobile Computing and Networking*, 2015, p. 155–166.

[99] X. Liu, Z. Zhou, W. Diao, Z. Li et K. Zhang, "When Good Becomes Evil: Keystroke Inference with Smartwatch," dans *22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, p. 1273–1285.

[100] Z. Liu, J. Liu, Y. Zeng et J. Ma, "Covert Wireless Communications in IoT Systems : Hiding Information in Interference," *IEEE Wireless Communications*, vol. 25, n°. 6, p. 46–52, déc. 2018.

[101] D. Konings, A. Budel, F. Alam et F. Noble, "Entity tracking within a Zigbee based smart home," dans *23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, nov. 2016, p. 1–6.

[102] D. Arp, E. Quiring, C. Wressnegger et K. Rieck, "Privacy threats through ultrasonic side channels on mobile devices," dans *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017, p. 35–47.

[103] W. Zhang, Y. Meng, Y. Liu, X. Zhang, Y. Zhang et H. Zhu, "HoMonit: Monitoring Smart Home Apps from Encrypted Traffic," dans *ACM SIGSAC Conference on Computer and Communications Security*, 2018, p. 1074–1088.

[104] Z. B. Celik, L. Babun, A. K. Sikder, H. Aksu, G. Tan, P. McDaniel et A. S. Uluagac, "Sensitive Information Tracking in Commodity IoT," 2018, p. 1687–1704.

[105] "Exodus Privacy."

[106] Z. Ma, H. Wang, Y. Guo et X. Chen, "LibRadar: Fast and Accurate Detection of Third-party Libraries in Android Apps," dans *38th International Conference on Software Engineering Companion*, 2016, p. 653–656.

[107] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun et H. Haddadi, "Information Exposure From Consumer IoT Devices : A Multidimensional, Network-Informed Measurement Approach," *Proceedings of the Internet Measurement Conference*, p. 13, 2019.

[108] O. Alrawi, C. Lever, M. Antonakakis et F. Monrose, "SoK : Security Evaluation of Home-Based IoT Deployments," *IEEE Symposium on Security and Privacy*, p. 19, 2019.

[109] N. Apthorpe, D. Reisman et N. Feamster, "Closing the Blinds : Four Strategies for Protecting Smart Home Privacy from Network Observers," *arXiv preprint arXiv :1705.06809*, 2017.

[110] N. Apthorpe, D. Y. Huang, D. Reisman, A. Narayanan et N. Feamster, "Keeping the Smart Home Private with Smart(er) IoT Traffic Shaping," *arXiv :1812.00955 [cs]*, déc. 2018.

[111] S. Nicolazzo, A. Nocera, D. Ursino et L. Virgili, "A privacy-preserving approach to prevent feature disclosure in an IoT scenario," *Future Generation Computer Systems*, vol. 105, p. 502–519, avr. 2020.

[112] H. Li, X. Lu, X. Liu, T. Xie, K. Bian, F. X. Lin, Q. Mei et F. Feng, "Characterizing Smartphone Usage Patterns from Millions of Android Users," dans *Internet Measurement Conference*, 2015, p. 459–472.

[113] E. J. Rader, "Awareness of Behavioral Tracking and Information Privacy Concern in Facebook and Google." dans *SOUPS*, 2014, p. 51–67.

[114] R. Kang, L. Dabbish, N. Fruchter et S. Kiesler, "My data just goes everywhere : user mental models of the Internet and implications for privacy and security," dans *Symposium on Usable Privacy and Security (SOUPS)*, 2015.

[115] P. Shayegh, V. Jain, A. Rabinia et S. Ghanavati, "Automated Approach to Improve IoT Privacy Policies," *arXiv :1910.04133 [cs]*, oct. 2019.

[116] B. Liu, M. S. Andersen, F. Schaub, H. Almuhimedi, S. A. Zhang, N. Sadeh, Y. Agarwal et A. Acquisti, "Follow my recommendations : A personalized privacy assistant for mobile app permissions," dans *Symposium on Usable Privacy and Security*, 2016.

[117] C. Stach et F. Steimle, "Recommender-based privacy requirements elicitation - EPICUREAN: an approach to simplify privacy settings in IoT applications with respect to the GDPR," dans *34th ACM/SIGAPP Symposium on Applied Computing - SAC '19*, 2019, p. 1500–1507.

[118] A. Pérez Fernández et G. Sindre, "Mitigating the Impact on Users' Privacy Caused by over Specifications in the Design of IoT Applications," *Sensors*, vol. 19, n°. 19, p. 4318, oct. 2019.

[119] H. Fereidooni, J. Classen, T. Spink, P. Patras, M. Miettinen, A.-R. Sadeghi, M. Hollick et M. Conti, "Breaking Fitness Records without Moving: Reverse Engineering and Spoofing Fitbit," *arXiv :1706.09165 [cs]*, juin 2017.

[120] P. Pongle et G. Chavan, "A survey : Attacks on RPL and 6LoWPAN in IoT," dans *International Conference on Pervasive Computing (ICPC)*, janv. 2015, p. 1–6.

[121] R. Krejčí, O. Hujňák et M. Švepeš, "Security survey of the IoT wireless protocols," dans *25th Telecommunication Forum*, nov. 2017, p. 1–4.

[122] T. Chothia et J. de Ruiter, "Learning From Others' Mistakes : Penetration Testing IoT Devices in the Classroom," dans *USENIX Workshop on Advances in Security Education (ASE 16)*. USENIX Association, 2016.

[123] "Les prochains ransomwares s'attaqueront aux objets connectés."

[124] J. Y. Kim, R. Holz, W. Hu et S. Jha, "Automated Analysis of Secure Internet of Things Protocols," dans *33rd Annual Computer Security Applications Conference (ACSAC)*, 2017, p. 238–249.

[125] B. Cyr, W. Horn, D. Miao et M. Specter, "Security analysis of wearable fitness devices (fitbit)," *Massachusets Institute of Technology*, p. 1, 2014.

[126] B. Michéle et A. Karpow, "Watch and be watched : Compromising all Smart TV generations," dans *IEEE 11th Consumer Communications and Networking Conference (CCNC)*, janv. 2014, p. 351–356.

[127] G. Chu, N. Apthorpe et N. Feamster, "Security and Privacy Analyses of Internet of Things Children's Toys," *IEEE Internet of Things Journal*, p. 1–1, 2018.

[128] M. Ikram, N. Vallina-Rodriguez, S. Seneviratne, M. A. Kaafar et V. Paxson, "An Analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps," dans *Proceedings of the 2016 ACM on Internet Measurement Conference*. ACM, 2016, p. 349–364.

[129] M. Vanhoef et F. Piessens, "Key reinstallation attacks : Forcing nonce reuse in WPA2," dans *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, p. 1313–1328.

[130] M. Ryan, "Bluetooth : With Low Energy Comes Low Security." *WOOT*, vol. 13, p. 4–4, 2013.

[131] X. Cao, D. M. Shila, Y. Cheng, Z. Yang, Y. Zhou et J. Chen, "Ghost-in-ZigBee : Energy Depletion Attack on ZigBee-Based Wireless Networks," *IEEE Internet of Things Journal*, vol. 3, n°. 5, p. 816–829, oct. 2016.

[132] E. Ronen, A. Shamir, A.-O. Weingarten et C. O'Flynn, "IoT goes nuclear : Creating a ZigBee chain reaction," 2017, p. 195–212.

[133] G. R, "La sécurité du protocole IoT Z-Wave est-elle compromise ?" mai 2018.

[134] G. Avoine et L. Ferreira, "Rescuing LoRaWAN 1.0," Rapport technique 651, 2017.

[135] N. Carlini et D. Wagner, "Audio Adversarial Examples : Targeted Attacks on Speech-to-Text," *arXiv preprint arXiv :1801.01944*, 2018.

[136] P. J. Young, J. H. Jin, S. Woo et D. H. Lee, "BadVoice : Soundless voice-control replay attack on modern smartphones," dans *Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, juill. 2016, p. 882–887.

[137] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang et W. Xu, "DolphinAtack: Inaudible Voice Commands," *arXiv :1708.09537 [cs]*, août 2017.

[138] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner et W. Zhou, "Hidden voice commands," dans *25th USENIX Security Symposium (USENIX Security 16), Austin, TX*, 2016.

[139] C. Kasmi et J. L. Esteves, "IEMI Threats for Information Security : Remote Command Injection on Modern Smartphones," *IEEE Transactions on Electromagnetic Compatibility*, vol. 57, n°. 6, p. 1752–1755, déc. 2015.

[140] W. Diao, X. Liu, Z. Zhou et K. Zhang, "Your Voice Assistant is Mine: How to Abuse Speakers to Steal Information and Control Your Phone," dans *4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, 2014, p. 63–74.

[141] J. Villalba et E. Lleida, "Preventing replay attacks on speaker verification systems," dans *2011 Carnahan Conference on Security Technology*, oct. 2011, p. 1–8.

[142] A. Apvrille et M. Pourzandi, "Secure Software Development by Example," *IEEE Security Privacy*, vol. 3, n°. 4, p. 10–17, juill. 2005.

[143] "New California Law Requires Strong Passwords for Internet of Things," 2018.

[144] "RGPD – Renforcer la sécurité des données à caractère personnel."

[145] X. Wang, Y. Sun, S. Nanda et X. Wang, "Looking from the Mirror: Evaluating IoT Device Security through Mobile Companion Apps," 2019, p. 1151–1167.

[146] E. Griffith, "If driverless cars save lives, where will we get organs?" août 2014.

[147] M. Thiery, V. Roca et A. Legout, "Privacy implications of switching ON a light bulb in the IoT world," juill. 2019.

[148] "Buy the Philips Hue White Single bulb E26 046677530341 Single bulb E26."

[149] "LED Bulb TRADFRI - IKEA."

[150] "LIFX Mini White."

[151] "Echo Spot | Alexa-enabled Speaker with 2.5" Screen - Black."

[152] "Google Home."

[153] "Gateway TRÅDFRI - IKEA."

[154] "Remote control TRÅDFRI - IKEA."

[155] "LG Nexus 5 - Full phone specifications."

[156] "Buy the Philips Hue Bridge 046677458478 Bridge."

[157] "Raspberry Pi 3 Model B."

[158] "Amazon Alexa – Applications sur Google Play."

[159] "all 4 hue – Applications sur Google Play."

[160] "Google Home – Applications sur Google Play."

[161] "Chrome : rapide et sécurisé – Applications sur Google Play."

[162] "Philips Hue – Applications sur Google Play."

[163] "Hue Hello (For Philips Hue Lights) – Applications sur Google Play."

[164] "IFTTT – Applications sur Google Play."

[165] "IKEA Home smart (TRÅDFRI) – Applications sur Google Play."

[166] "LIFX - Apps on Google Play."

[167] "openHAB – Applications sur Google Play."

[168] "openHAB."

[169] "home-assistant/home-assistant," nov. 2019.

[170] "Python Data Analysis Library — pandas: Python Data Analysis Library."

[171] P. Cipolloni, "Universal Android SSL Pinning bypass with Frida | @Mediaservice.net Technical Blog."

[172] "Security Analysis for IoT devices | Completely Automated."

[173] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman et V. Paxson, "Haystack: A Multi-Purpose Mobile Vantage Point in User Space," *arXiv preprint arXiv :1510.01419*, 2015.

[174] "Ma Deuce."

[175] "Chromania."

[176] T. E. Parliament et the Council of the European Union, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," 2016.

[177] E. D. P. Board, "Guidelines on transparency under regulation 2016/679 (wp260rev.01)."

[178] "Règles de confidentialité Google," oct 2019.

[179] "Notice : Protection de vos informations personnelles," sep 2019.

[180] "Privacy Policy – LIFX Europe," aug 2019.

[181] "Privacy Policy – IFTTT," jul 2018.

[182] A. M. McDonald et L. F. Cranor, "The cost of reading privacy policies," *ISJLP*, 2008.

[183] J. A. Obar et A. Oeldorf-Hirsch, "The biggest lie on the internet : Ignoring the privacy policies and terms of service policies of social networking services," *Information, Communication & Society*, p. 1–20, 2018.

[184] "The CNIL's restricted committee imposes a financial penalty of 50 Million euros against GOOGLE LLC," 2019.

[185] A. Mathur, G. Acar, M. Friedman, E. Lucherini, J. R. Mayer, M. Chetty et A. Narayanan, "Dark patterns at scale : Findings from a crawl of 11k shopping websites," *Proceedings of the ACM Human-Computer Interaction*, vol. 3, 2019.

[186] E. D. P. Board, "Guidelines on consent under regulation 2016/679 (wp259rev.01)."

[187] ——, "Eu–u.s. privacy shield – third annual joint review report," 2019.

[188] V. Morel, "Enhancing transparency and consent in the iot," 2020.

[189] K. Hinton, J. Baliga, M. Feng, R. Ayre et R. S. Tucker, "Power consumption and energy efficiency in the internet," *IEEE Network*, vol. 25, n°. 2, p. 6–12, mars 2011.

[190] A. Vishwanath, F. Jalali, K. Hinton, T. Alpcan, R. W. A. Ayre et R. S. Tucker, "Energy Consumption Comparison of Interactive Cloud-Based and Local Applications," *IEEE Journal on Selected Areas in Communications*, vol. 33, n°. 4, p. 616–626, avr. 2015.