Solutions de sécurité pour l'internet des objets dans le cadre de l'assistance à l'autonomie à domicile

Fatma Merabet

THÈSE EN COTUTELLE

pour obtenir le grade de Docteur délivré par

L'Université Mouloud Mammeri de Tizi Ouzou et de L'Université de Limoges

Spécialité doctorale "Mathématique et Informatique"

Option "Réseaux, Mobilité et Systèmes Embarqués"

présentée et soutenue publiquement par

Fatma MERABET

le 01/06/2021

Solutions de Sécurité pour l'Internet des Objets dans le Cadre de l'Assistance à l'Autonomie à Domicile

Sous la codirection de

BELKADI Malika,	Professeur	Université Mouloud Mammeri de Tizi Ouzou
SAUVERON Damien,	M.C/HDR	Université de Limoges

Jury

LALAM Mustapha,	Professeur	Université Mouloud Mammeri de Tizi Ouzou	Président
ALI PACHA Adda,	Professeur	Université d'Oran	Examinateur
PEYRARD Fabrice,	M.C/HDR	Institut de Recherche en Informatique de Toulouse	Examinateur
CONCHON Emmanuel,	M.C/A	Université de Limoges	Invité

Remerciements

Ce travail a été réalisé au sein du laboratoire de recherche en informatique (LARI) de l'Université Mouloud Mammeri de Tizi-Ouzou et du laboratoire (XLIM) de l'Université de Limoges (France).

J'adresse tous mes remerciements à mes directeurs de thèse Malika BELKADI et Damien SAUVERON, et aussi ma profonde gratitude à mon codirecteur de thèse Emmanuel CONCHON. Je les remercie pour leur attention, leurs conseils avisés et leur contribution à la réussite de cette thèse. Je suis ravie d'avoir travaillé en leur compagnie.

Je remercie le directeur du laboratoire de recherche (LARI) monsieur Mustapha LALAM, Professeur à l'Université Mouloud Mammeri de Tizi-Ouzou, pour avoir accepté de présider mon jury. J'associe à ces remerciements tous les membres du laboratoire LARI ainsi que tous les enseignants du département d'informatique et les membres du laboratoire XLIM.

J'adresse mes remerciements à monsieur Adda ALI PACHA, Maître de Conférences à l'Université d'Oran, à monsieur Fabrice PEYRARD, Maître de Conférences à l'Institut de Recherche en Informatique de Toulouse et à monsieur Emmanuel CONCHON, Maître de Conférences Université de Limoges qui vont me faire l'honneur de juger mon travail.

Je ne pourrais pas finir ces remerciements sans en adresser les plus chaleureux à mes parents et à toute ma famille pour leur soutien continu et leurs encouragements.

Résumé

L'Internet des Objets (IdO ou IoT en anglais) est un paradigme présent dans la vie quotidienne de chaque être humain. En effet, l'IdO comprend aujourd'hui une très grande diversité d'objets intégrants des capteurs, des actionneurs et également des modules de communication leur permettant de s'intégrer à Internet. Ces dispositifs observent et agissent sur l'état du monde réel constituant ainsi une passerelle entre la vie réelle et le monde numérique.

L'IdO fournit de nombreux services avancés comme la surveillance automatique en temps réel d'environnements industriels, médicaux, agricoles, et autres qui peuvent être couverts et gérés automatiquement. L'Assistance à l'Autonomie des personnes à Domicile (AAD ou AAL) est l'un des nombreux domaines bénéficiant des avancées permises par l'IdO. L'AAD vise à aider les personnes âgées, malades ou dépendantes dans leurs routines quotidiennes en fournissant de nouveaux services d'assistance dans les maisons intelligentes (« smart home ») afin de surveiller en continue leurs constantes physiologiques à moindre coût pour détecter de possibles dégradations de leur état de santé. Cependant, malgré tous les avantages qu'apporte l'IdO dans la vie quotidienne de ces personnes dépendantes, les problématiques de sécurité freinent considérablement le déploiement de ces technologies. En effet, pour permettre une plus large adoption des solutions d'AAD s'appuyant sur l'IdO, la sécurité des objets et des réseaux utilisés doit être renforcée. Ainsi, de nouveaux mécanismes légers et robustes doivent être développés pour contrer les menaces et les attaques auxquelles les infrastructures de l'IdO sont exposées. Une première ligne de défense est d'utiliser des mécanismes d'authentification adaptés aux capacités et aux contraintes des objets et des technologies de communication afin d'assurer les services d'authentification des objets et de mettre en œuvre des canaux sécurisés permettant la protection des données.

Dans cette thèse, notre objectif est de proposer des solutions de sécurité pour l'IdO dans le cadre médical. Pour cela, nous dressons tout d'abord un état de l'art de l'IdO. Puis, nous présentons les défis de sécurité auxquels l'IdO est confronté. Nous considérons alors les protocoles d'authentification destinés à garantir la sécurité et le respect de la vie privée des utilisateurs des objets connectés. Ainsi, notre première contribution consiste en la proposition de trois nouveaux protocoles d'authentification mutuelle pour les objets connectés avec deux modes de communication

différents M2M et M2C, dans le cadre de l'assistance à l'autonomie des personnes à domicile. En plus de l'analyse informelle une validation formelle de sécurité de nos propositions a été menée en utilisant les outils AVISPA et ProVerif. Notre deuxième contribution consiste en un nouveau protocole d'identification et d'authentification pour l'AAD, adapté à la technologie RFID. Ce protocole vise à assurer l'authentification entre un lecteur et des étiquettes RFID pour garantir la sécurité et le respect de la vie privée des personnes dépendantes.

Mots clés : *IoT, AAL, WSN, RFID, Protocoles d'authentification, e-santé.*

Abstract

Nowdays, the Internet of Things (IoT) represents a major part of our daily lives. IoT includes a wide variety of objects integrating sensors and communication modules that facilitate their integration on the internet. These devices observe and measure the state of the real world and connect real life with the virtual world.

IoT provides advanced services such as real-time monitoring of environments as industrial, medical, agricultural, and other environments that can be automatically covered and managed. Ambient Assisted Living (AAL) is one of many areas that benefit from IoT. It aims to help older people in their daily routines by providing new assistance services in smart homes to monitor their physiological constants at lower cost to detect possible health degradations. However, despite all the benefits that IoT brings in our daily lives, some may be a real inconvenience. Security issues greatly impede the development and rapid deployment of this technology. The widespread of IoT can only be successful when adequate security of the objects and the used communication networks is adapted.

New lightweight and robust mechanisms need to be adapted to counter the threats and attacks to which IoT infrastructure communications are exposed. A first often a prerequisite mandatory step would be the use of authentication mechanisms that provide the services of object authentication and data protection, while being adapted to the capabilities and respecting the limits of objects and communication technologies.

In this thesis, our goal is to implement security solutions for IoT in the medical field. For this, we first establish the state of the art of IoT. Subsequently, we address the security challenges faced by the evolution of IoT. We also consider authentication protocols to ensure the security and privacy of users of connected objects. Afterward, we present a first contribution which consists in three new mutual authentication protocols for connected objects with two different communication modes Machine-to-Machine (M2M) and Machine-to-Cloud (M2C), as part of AAL. Validation of security of our proposals was made using the tools AVISPA and ProVerif in addition to the informal analysis. Our second contribution includes a new identification and authentication protocol in the field of RFID technologies to ensure authentication between a reader and RFID tags in order

to guarantee patients security and privacy, as part of an RFID medical assistive environment.

keywords : *IoT, AAL, WSN, RFID, Authentication Protocol, Healthcare.*

Table des matières

Liste des figures

Liste des tableaux

Glossaire

AAD Assistance à l'Autonomie des personnes à Domicile, aussi nommé AAL en anglais.

AAL « Ambient Assisted Living ».

ABE « Attribute Based Encryption ».

ADR « Adaptive Data Rate ».

AES « Advanced Encryption Standard ».

BAN « Body Area Network ».

BASN « Body Area Sensor Network ».

BCU « Body Central Unit ».

BLE « Bluetooth Low Energy ».

BSN « Body Sensor Network ».

CA « Certificate Authority ».

CGC Centre Génération des Clés.

CoRE « Constrained Resource Environments ».

CPABE « Ciphertext-Policy Attribute Based Encryption ».

CPL Courants Porteurs en Ligne.

CRHF « Collision Resistant Hash Function ».

DH « Diffie–Hellman ».

DoS « Deny of Service ».

DTLS « Datagram Transport Layer Security ».

E2E « End-to-End ».

ECC « Elliptic Curve Crypthography ».

ECDSA « Elliptic Curve Digital Signature Algorithm ».

ECG « Électrocardiographie » aussi nommé Electrocardiogram en anglais .

EEG « Électroencéphalogramme » aussi nommé Electroencephalogram en anglais .

EMR Electronic Medical Record.

EPC « Electronic Product Code ».

FE « Functional Encryption ».

GPS « Global Positioning System ».

HAN « Home Area Network ».

HLPSL « High Level Protocol Specification Language ».

HTTP « HyperText Transfer Protocol ».

IBM « International Business Machines ».

ICC « Integrated Chip Circuit ».

ICDs « Implantable Cardiac Defibrillators ».

IdO Internet des Objets, aussi nommé IoT en anglais.

IETF « Internet Engineering Task Force ».

IIoT « Industrial IoT ».

IoT « Internet of Things ».

LoRaWan « Long Range Wide-area networK ».

LPL « Low Power Listening ».

M2C « Machine-to-Cloud ».

M2M « Machine-to-Machine ».

MAC « Message Authentication Code ».

MDC « Manipulation Detection Code ».

MITM « Man in the Middlet ».

MTU « Maximum Transmission Unit ».

NFC « Near Field Communication ».

OWHF « One-Way Hash Functionr ».

PC « Personnel Computer ».

PRE « Proxy Re-Encryption ».

RFID « identification par radio-fréquence » aussi nommé Radio Frequency IDentification en anglais.

RPL « Routing Protocol for Low-Power and Lossy Networks ».

RSA « Rivest-Shamir-Adleman ».

SHA « Secure Hash Algorithm ».

SIG « Special Interest Group ».

SSL « Secure Socket Layer ».

TLS « Transport Layer Security ».

UNB « Ultra Narrow Band ».

WBAN « Wireless Body Area Network ».

WEP « Wired Equivalent Privacy ».

Wi-Fi « Wireless-Fidelity ».

WLAN « Wireless Local Area Network ».

WMSN « Wireless Medical Sensor Network ».

WPA « Wi-Fi protected access ».

WSN « Wireless Sensor Networks ».

WWW « Word Wide Web ».

Introduction générale

Contexte et motivation

Au cours des dernières années grâce à la miniaturisation toujours plus poussée des composants électroniques et aux innovations technologiques en matière de communication numérique, le nombre d'objets intelligents connectés n'a pas cessé d'augmenter dans tous les domaines d'applications.

Par définition, un objet intelligent est une machine physique ou virtuelle qui doit avoir une certaine capacité de calcul et de mémorisation en plus d'être autonome ; c'est-à-dire qu'il peut faire des traitements et parfois même prendre des décisions sans intervention humaine. Mais aussi, il peut être connecté avec n'importe quel autre objet d'une manière flexible et transparente. L'ensemble de ces objets connectés, formant l'Internet des Objets (IdO) ou « Internet of Things » (IoT) en anglais, peuvent assister les humains dans les activités de la vie quotidienne grâce à leurs capacités de détection, de calcul et de communication.

L'utilisation de ces dispositifs connectés dans des environnements perceptifs concernant des lieux de vie comme les maisons intelligentes, offre de nouvelles perspectives pour améliorer la qualité de vie des personnes et trouve une application directe dans le cadre de l'assistance à l'autonomie à domicile (AAD). En effet, avec le vieillissement des personnes dans les pays développés, les maisons intelligentes sont une des solutions envisagées pour permettre aux personnes âgées de rester à leur domicile plus longtemps. Ce type de maisons apporte également de nombreux avantages aux personnes à mobilité réduite en leur permettant une plus grande autonomie, améliorant ainsi leur confort de vie. L'utilisation d'équipements d'auto-mesures est aussi une solution permettant de surveiller certaines constantes physiologiques de personnes dépendantes pour obtenir des informations suffisamment précises sur leur état de santé afin de détecter de possibles dégradations et d'assurer au mieux leur prise en charge par les services médicaux.

Toutefois, la prolifération de ces objets connectés qui manipulent des données personnelles sensibles met en péril la vie privée de leurs utilisateurs. En effet, ces informations relatives aux habitudes, aux maladies, à la situation des personnes pourraient être utilisées par des individus non autorisés désirant en tirer un profit illicite. C'est pourquoi, la sécurité et la confidentialité des

données des utilisateurs représentent des défis importants pour l'IoT dans le domaine de la santé et en particulier dans celui de l'AAD. De plus, l'utilisation de l'Internet des Objets introduit des défis pour la mise en œuvre d'outils de sécurité traditionnels. En effet, du côté de l'IoT, la rareté des ressources en termes d'énergie, de mémoire et de capacité de calcul, qui caractérise les objets connectés, limite le déploiement des solutions de sécurité traditionnelles. Une première exigence serait d'utiliser des mécanismes d'authentification comme première barrière aux menaces que rencontrent les communications de l'IoT pour garantir la légitimité des entités communicantes. Ces mécanismes d'authentification doivent reposer sur des algorithmes de chiffrement plus légers et adaptés aux ressources des objets connectés utilisés.

Dans cette thèse, notre but est de proposer des solutions de sécurité pour l'Internet des Objets utilisé dans le domaine médical et en particulier dans l'AAD. Nous visons à sécuriser les dispositifs à faible capacité de calcul qui sont utilisés par les patients ou présents dans leur environnement et qui accèdent à leur données médicales et personnelles. Notamment, l'utilisation des protocoles de sécurité dans le cadre des réseaux domiciliaires (Home Area Network) permet de sécuriser les échanges de données tout en respectant la vie privée des patients qui occupent le domicile.

Objectifs de la thèse

Afin de faire face aux problèmes et défis susmentionnés, nous nous sommes fixés les objectifs suivants :

— **Objectif A :** (1) Établir un état de l'art sur l'Internet des Objets et ses différentes technologies, les défis à relever pour accompagner son évolution. (2) Étudier son utilisation dans le domaine médical.

— **Objectif B :** (1) Étudier les problématiques de sécurité de l'Internet des Objets en général et les solutions proposées dans la littérature. (2) Faire une étude approfondie sur les mécanismes de sécurité utilisés dans l'Internet des Objets en général et leur application dans le domaine médical.

— **Objectif C :** Proposer des protocoles de sécurité adaptés aux ressources limitées des objets connectés du domaine de la santé et permettant d'assurer la confidentialité, l'intégrité et le respect de la vie privée des utilisateurs.

— **Objectif D :** Présenter des analyses de sécurité à la fois informelles des protocoles proposés, mais aussi prouver avec une analyse formelle qu'ils satisfont réellement les propriétés de sécurité attendues.

— **Objectif E :** Procéder à une évaluation expérimentale des performances de protocoles proposés implémentés sur des dispositifs sécurisés contraints en ressources.

Organisation de la thèse

Cette thèse est organisée comme suit :

— **Chapitre 1 :** Dans ce chapitre nous présentons un état de l'art de l'Internet des Objets. Nous étudierons donc son évolution et les défis à relever pour faciliter et augmenter son adoption, les protocoles et les technologies de communication de l'IoT. En particulier, nous nous concentrerons sur ceux pouvant être utilisés pour connecter tous les dispositifs IoTs dans les maisons intelligentes. Nous introduirons aussi l'application de l'IoT dans le domaine médical et présenterons ses avantages dans le cadre de l'assistance à l'autonomie à domicile. Ce chapitre permet de répondre à l'**Objectif A**.

— **Chapitre 2 :** Ce chapitre est consacré à la sécurité dans l'Internet des Objets. Pour atteindre l'**Objectif B**, nous étudions les problèmes de sécurité dans la technologie de l'IoT, les besoin de sécurité qui doivent être garantis pour les contrer. En se basant sur ces études nous tirons des conclusions sur le besoin d'authentification qui est primordial pour garantir l'authenticité des entités communicantes.

— **Chapitre 3 :** Dans ce chapitre, nous présentons trois nouveaux protocoles d'authentification qui consistent à garantir la confidentialité, l'intégrité et le respect de la vie privée des entités communicantes que ce soit entre un objet et le serveur ou entre un objet et un autre objet, dans le cadre de l'assistance à la vie ambiante. Nous commençons par la description du scénario utilisé puis nous détaillons les étapes des protocoles. Nous analysons ensuite leur sécurité en se basant sur le modèle d'attaquant et nous procédons à une vérification formelle à l'aide des outils AVISPA et ProVerif. Par la suite, nous présentons les résultats de l'implémentation des protocoles sur des dispositifs très limités en ressources (Universal JCard et Multos IoT Trust Anchor) pour évaluer son adéquation aux exigences de performance. Ce chapitre permet de répondre aux **Objectifs C, D et E**.

— **Chapitre 4 :** Dans ce chapitre, nous présentons une autre contribution qui consiste en un protocole d'authentification pour des technologies radio-fréquences (RFID) entre un lecteur et des étiquettes RFID permettant de garantir la sécurité et le respect de la vie privée des patients. Ce protocole repose uniquement sur des fonctions de hachage unidirectionnelles sans collision et sur de l'identification de groupe ce qui le rend plus léger que les solutions présentées au chapitre précédent. Nous commençons par présenter le cadre général d'application de ce protocole avec les exigences de sécurité afférentes ainsi que le modèle d'attaquant considéré. Après un état de l'art centré sur les solutions légères d'authentification pour les réseaux RFID, nous étudions notre proposition puis nous analysons ensuite sa sécurité à travers une analyse informelle et une vérification formelle à l'aide des outils

AVISPA et ProVerif. Enfin, une évaluation de performance théorique est réalisée. Ce chapitre permet de répondre aux **Objectifs C et D**.

— **Chapitre 5 :** Dans ce chapitre, nous présentons une dernière contribution qui est en cours de finalisation et qui propose un protocole de communication sécurisée permettant la collecte des données de capteurs présents dans un domicile et sur un patient afin de permettre une assistance médical. Après une présentation des réseaux WBAN au sein du domicile et un tour d'horizon de solutions concurrentes, nous développons notre proposition qui s'inspire de travaux existants mais pour lesquels malgré l'existence de preuves formelles, il existait des attaques que nous corrigeons tout en conservant une complexité calculatoire très raisonnable et adaptée aux dispositifs IoT. Ce chapitre permet de répondre aux **Objectifs C et D**.

Chapitre 1

L'Internet des Objets et son application dans le domaine médical

Sommaire

1.1 Introduction

Avec l'évolution rapide des différentes technologies de communication sans fil et leur intégration dans la plupart des objets intelligents, notre vie quotidienne va subir des changements dans de nombreux domaines. Ces objets intelligents sont souvent des capteurs dotés de capacités de mesures (température, pression, vibration, luminosité, humidité, tension, etc.) ou des actionneurs capables d'agir. Ces objets disposent de différentes technologies de communication : RFID (« Radio Frequency IDentification ») [4], NFC (« Near Field Communication ») [5], Bluetooth [6], Wi-Fi [7], LoRa [8], etc. et ainsi, ils peuvent s'interconnecter pour anticiper et interagir en temps réel, c'est-à-dire pour réaliser un objectif commun (surveillance de l'environnement, contrôle du trafic routier urbain, etc.). Ensemble, tous ces objets connectés constituent la base du paradigme de l'Internet des Objets [9] qui permet aux personnes/objets d'être connectés à tout moment et en tout lieu [10].

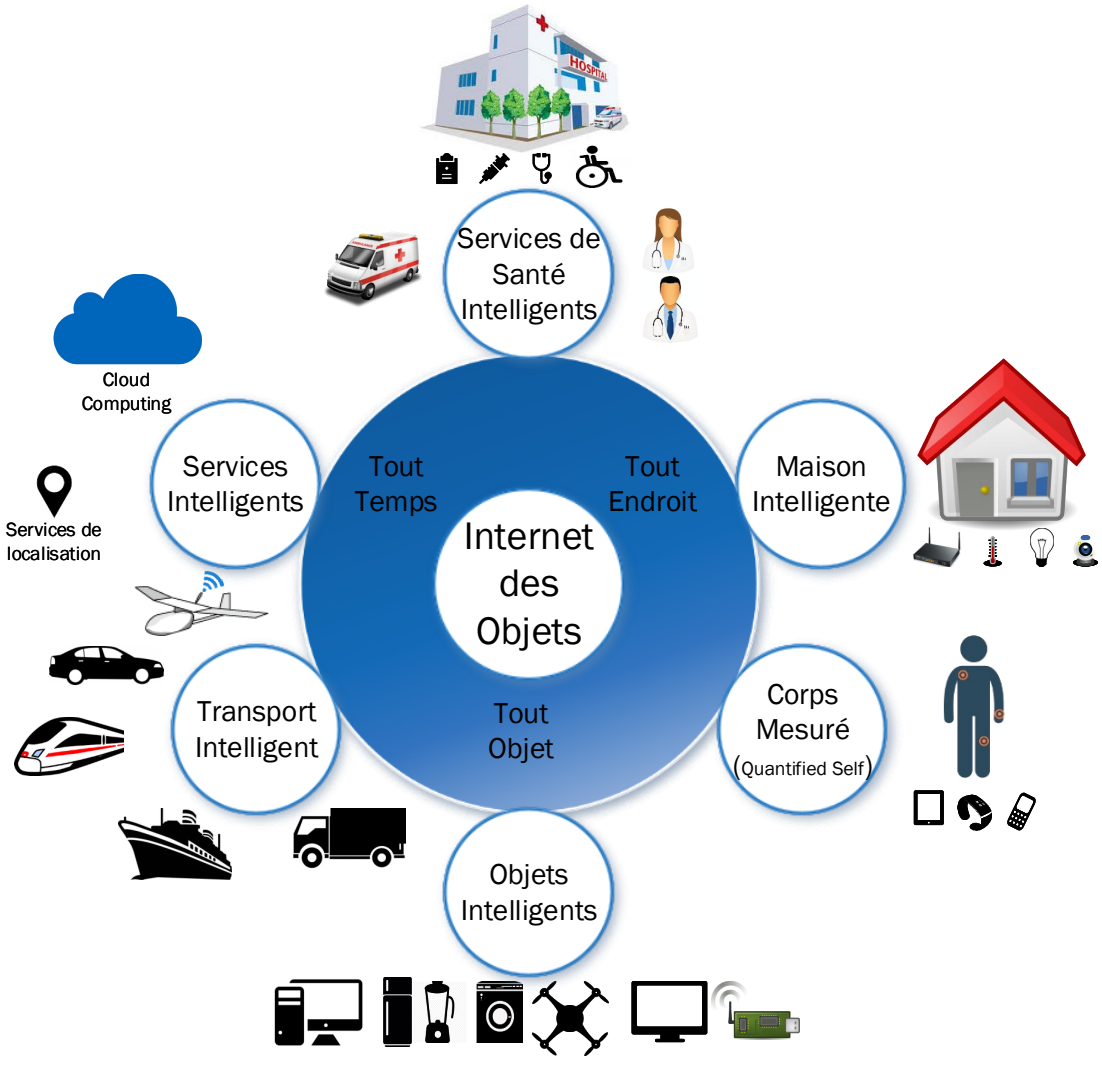


FIGURE 1.1 – Paradigme de l'Internet des Objets

L'IoT comprend des objets physiques équipés de dispositifs intelligents pour former des objets intelligents (que l'on nomme également dispositifs/objets connectés dans la suite du manuscrit) qui communiquent entre eux via le réseau et recueillent des données accessibles à partir de tout endroit dans le monde. L'IoT se caractérise par l'utilisation de technologies de communication hétérogènes (LoRa, SigFox, etc.) qui collaborent pour fournir des services innovants dans divers domaines d'application [11, 12]. Son but est au travers de modes de communication avancés entre les objets intelligents et les systèmes, de faciliter les interactions des humains avec le monde virtuel.

Parmi les différents modes de communication entre objets connectés, on trouve essentiellement le Machine à Machine M2M (« Machine-to-Machine ») et le Machine à Cloud M2C (« Machine-to-Cloud »). Le type de communication M2M permet aux objets intelligents d'échanger et de partager des données de manière décentralisée indépendamment d'un système centralisé [13, 14, 15]. À l'inverse le type de communication M2C est centralisé entre les objets intelligents et le cloud.

Si l'Internet des Objets peut s'appliquer à plusieurs domaines (le transport intelligent, les services de santé intelligents, les maisons intelligentes, les villes intelligentes, etc. comme le montre la Figure 1.1) dans cette thèse c'est davantage le domaine de la santé qui nous intéressera. Par exemple, dans le domaine médical des patients intègrent des dispositifs intelligents qui offrent aux services de santé la possibilité de diagnostiquer et de déterminer s'ils sont sujets à certaines pathologies [16, 17]. Pour les hôpitaux aussi, l'IoT et le partage de données sont également des vecteurs de gains économiques (centralisation et accès direct aux données via le cloud, mise à jour des informations médicales en continu, etc.) [18].
Les données relevées en temps réel par les dispositifs portés sont des indicateurs de l'état de santé de son propriétaire. Par exemple, les capteurs de mouvements intégrés d'une montre intelligente peuvent être utilisés pour suivre les symptômes des maladies telles que les tremblements chez les patients atteints de la maladie de Parkinson [19]. De plus, les dispositifs portés sont utilisés pour l'autosurveillance et pour la prévention des problèmes de santé tels que l'hypertension et le stress.

La prolifération de ces objets connectés à travers les solutions domotiques peut également apporter beaucoup d'avantages aux personnes à mobilité réduite telles que les personnes âgées ou handicapées. En effet, cela leur offre une plus grande autonomie et améliore leur confort dans le cadre de l'AAD (« Assistance à l'Autonomie à Domicile ») ou, en anglais, AAL (« Ambient Assisted Living ») [20].

Par ailleurs, dans le domaine de la santé, l'IoT facilite le déploiement de réseaux personnels pour le contrôle et le suivi des données cliniques, notamment pour des personnes âgées. Il permet aussi de faciliter le suivi des patients à domicile et d'apporter des solutions pour améliorer l'autonomie des personnes à mobilité réduite. Ainsi, grâce à l'ubiquité des systèmes de commu-

nication permettant à la relation docteur-patient d'être entretenue à distance, les maisons intelligentes deviennent une part importante du système de santé, et l'hôpital reste un environnement de spécialité et le lieu des soins cruciaux. Demain, de tels scénarios pourraient améliorer la qualité des soins médicaux reçus par un patient, et en situation d'urgence, réduire le temps de transit entre ambulances et salles d'urgence et ainsi réduire les complications pouvant survenir lors de ces transitions [21, 22]. D'une façon générale, dans le domaine médical, les objets connectés sont synonymes de gain de temps et de précision dans la prise en charge des patients ; ils permettent ainsi de prévenir les risques et d'ajuster les traitements par le médecin traitant en cas de nécessité.

L'objectif de ce premier chapitre est de présenter une vue générale de l'Internet des Objets et son application dans le domaine médical. En premier lieu, nous allons introduire l'évolution de l'IoT et les défis auxquels il est confronté. Ensuite, nous allons décrire les principales technologies de communication utilisées par les objets intelligents, puis, les deux technologies clés de l'IoT que sont RFID et WSN. Pour finir, nous nous intéresserons à l'utilisation des technologies de l'IoT dans le domaine de la santé et des différents services qu'elles peuvent offrir dans le cadre de AAD pour les personnes âgées et les personnes à activités réduites. Dans ce contexte, nous présenterons un modèle d'architecture dans le cadre de ces applications permettant de mettre en exergue les modes de communication M2M et M2C pour lesquels nous ferons des propositions de sécurisation dans la suite du manuscrit.

1.2 L'évolution de l'Internet des Objets

L'Internet des Objets est en évolution permanente. Aujourd'hui, l'IoT a pour ambition de rendre intelligents les objets de notre quotidien en leur offrant la faculté de s'interconnecter et d'interagir entre eux, avec leur environnement physique ou avec le nuage (« cloud »). Aussi, pour comprendre son évolution, dans la suite, nous commençons par étudier les fondements du concept d'IoT, pour ensuite aborder son utilisation, et donc in fine celle des objets connectés qui le composent, dans les principaux concepts utilisés dans cette thèse (de l'informatique ubiquitaire à l'informatique dans le nuage).

1.2.1 De l'Internet à l'Internet des Objets

Dans les années 1960, l'apparition du réseau Internet a révolutionné les communications des réseaux informatiques traditionnels (d'ordinateurs à ordinateurs) en ouvrant la possibilité de collaboration et d'interaction entre les individus et leurs ordinateurs à l'échelle mondiale. L'introduction du protocole TCP/IP en 1980 et l'apparition de WWW (« Word Wide Web ») en 1991 ont grandement contribué à populariser Internet.

Dans l'Internet des Objets, l'ordinateur n'est plus le seul objet capable de communiquer et, au contraire, tout objet non dédié à l'informatique (un téléphone « smartphone », une tablette, une voiture, une télévision, etc.) peut échanger de l'information. Un des premiers dispositifs non informatique connecté à Internet fut un grille-pain [1] pouvant être activé et désactivé au travers du réseau Internet; il a été créé en 1990 par John Romkey. Steve Mann, quant à lui, a inventé en 1994 WearCam [2] une caméra portable connectée sans fil. En 1997, Paul Safo a fait la première description de ce que l'on nomme aujourd'hui *capteurs* dans son article « Sensors : The Next Wave of Infotech Innovation » [3] [23]. Ce n'est qu'en 1999 que le terme IoT (« Internet of Things ») a été utilisé pour la première fois par le britannique Kevin Ashton pour décrire un système dans lequel les objets dans le monde physique pourraient être connectés à l'Internet en utilisant une des premières technologies sans fil : la RFID (« Radio Frequency IDentification »). La même année, un standard d'identification des produits basés sur la technologie RFID, EPC (« Electronic Product Code »), a été inventé pour remplacer les code-barres. Rapidement les grands industriels saisissent l'opportunité et dès les années 2000, de premiers produits connectés grand public sont apparus. Ainsi, le fabricant LG annonçait le premier réfrigérateur connecté « LG Internet Digital DIOS » capable par exemple de détecter la présence d'aliments à l'intérieur et de garder une traçabilité du stock grâce à la lecture de codes à barres ou d'identifiants RFID [24]. En 2011, le nombre de dispositifs interconnectés dépassait la population mondiale et il devait atteindre les 50 milliards de dispositifs en 2020 [25, 26, 9]. Le concept d'IoT a donc évolué au fil du temps : il a connu plusieurs transformations successives lors de ces trois dernières décennies, et encore aujourd'hui on voit poindre des évolutions pour les prochaines années avec la miniaturisation, l'intégration de nouvelles technologies de communication, la massification des objets et leur adoption dans différents domaines d'application.

1.2.2 L'informatique ubiquitaire

Au début de l'apparition et de l'utilisation des ordinateurs, plusieurs personnes partageaient le même ordinateur. Puis, grâce à la miniaturisation des systèmes électroniques et à la réduction des coûts, les ordinateurs personnels (aussi appelés PCs pour « Personnel Computers ») sont apparus. Ils étaient alors principalement détenus et utilisés par une seule personne à laquelle ils étaient dédiés [27].

De par les avancées technologiques récentes dans les domaines des technologies de l'information et des communications en lien avec la miniaturisation des dispositifs (capteurs, processeurs, actionneurs), mais aussi les évolutions dans les usages avec le développement du nomadisme

1. https ://www.postscapes.com/internet-of-things-history/
2. http ://wearcam.org/myview.html
3. http ://www.saffo.com/essays/sensors-the-next-wave-of-infotech-innovation/

grâce à la mobilité des équipements et l'utilisation des communications sans fil, les systèmes informatiques actuels ne sont plus constitués seulement d'ordinateurs connectés à des serveurs. Ils intègrent en effet une grande variété de nouveaux dispositifs communiquants embarqués qui posent les bases conceptuelles de systèmes informatiques ubiquitaires. L'informatique ubiquitaire (« ubiquitous computing »), appelée également informatique omniprésente et invisible, ou encore parfois, intelligence ambiante (« ambient intelligence »), est un paradigme consistant à introduire de manière transparente les technologies informatiques au quotidien de l'Homme. Aujourd'hui, les utilisateurs possèdent en effet un nombre croissant de dispositifs connectés intégrés à divers objets physiques, tels que les véhicules modernes, les appareils électroménagers, les dispositifs médicaux implantés, qui participent à ce qu'on nomme l'IoT. En substance, l'informatique ubiquitaire combine les objets du monde réel à ceux du monde virtuel.

1.2.3 La domotique

La domotique vise à automatiser et/ou à contrôler localement et/ou à distance les solutions techniques permettant de répondre aux besoins de confort dans l'habitat, comme la gestion de l'énergie, de l'éclairage et du chauffage, la protection du lieu de vie par l'installation et la maintenance de caméras de surveillance des alarmes, etc. Autrement dit, la domotique regroupe l'ensemble des solutions d'électroniques, d'automatisme, de communication et d'informatique pour le contrôle et la gestion centralisée d'appareils électriques et électroniques dans tous bâtiments (hôtels, établissements recevant du publique comme les laboratoires, les entreprises) et surtout historiquement dans le domicile. Dans le cadre domiciliaire, le concept du bien-être chez soi a tellement été développé que l'on parle maintenant de maisons intelligentes.

1.2.4 Les maisons intelligentes (« smart homes »)

Le concept de maison intelligente est né à la fois de la rencontre des nouveaux concepts d'informatique ubiquitaire et de domotique mais aussi du contexte favorable du domicile qui constitue un domaine d'application à la pointe pour explorer leurs potentiels d'innovation.

Effectivement, les technologies de la maison intelligente comprennent des capteurs, des actionneurs, des appareils et des périphériques mis en réseau afin de permettre l'automatisation ainsi que le contrôle locale et à distance de l'environnement domestique. Les dispositifs contrôlables comprennent les systèmes de chauffage et d'eau chaude (chaudières, radiateurs), l'éclairage, les fenêtres, les rideaux, les portes de garage, les réfrigérateurs, les téléviseurs et les machines à laver. De ce fait, en plus des tâches habituelles pour lesquelles ils ont été conçus, il est possible d'y avoir accès et de pouvoir les commander à distance (localement ou par internet via différentes technologies de communication). Ces dispositifs agissent ainsi comme des points d'ac-

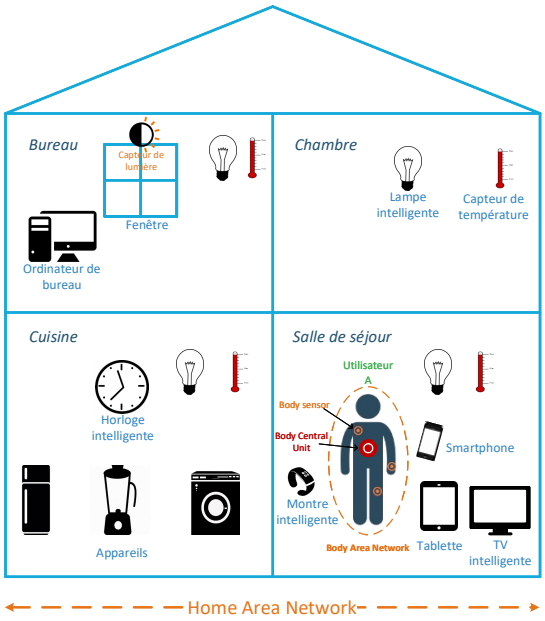


FIGURE 1.2 – Maison intelligente

cès offrant différents services. Les capteurs détectent les paramètres environnementaux tels que la température, la lumière, les mouvements et l'humidité. Un exemple de services d'une maison intelligente est la récupération d'informations à partir de capteurs pour commander à des actionneurs d'exécuter des actions spécifiques. Par exemple, dans une maison intelligente, il est naturel d'avoir un ou plusieurs capteurs de température qui communiquent avec le chauffage central de la maison pour conserver une température ambiante spécifique dans la maison, voire dans chaque pièce. Autrement dit, ces objets intelligents complètent les installations existantes en augmentant le bien-être, et ainsi ils participent à la réalisation des maisons connectées comme le montre la Figure 1.2.

La possibilité de contrôle est fournie soit par un logiciel installé sur les périphériques informatiques (téléphone portables, tablettes, ordinateurs portables, ordinateurs personnels, etc.) soit par des interfaces matérielles dédiées (par exemple, écrans de contrôle et de commande installés sur les murs). Les différents dispositifs sont mis en réseau, généralement via une technologie sans fil, à l'aide de protocoles de communication normalisés (section 1.4). La diversité des dispositifs et des technologies offre une grande souplesse dans leurs déploiements dans les maisons intelligentes. L'intelligence de ces nouvelles maisons réside dans la multitude des nouveaux services et dans leur agencement [28].

1.2.5 L'informatique en nuage (« cloud computing »)

Avec la démocratisation de l'IoT, les objets intelligents sont devenus omniprésents dans le quotidien. Pour de nombreux services nécessitant l'implication de différents objets, ces derniers

doivent communiquer entre eux pour échanger des informations. Toutefois, devant les grandes quantités de données qu'ils génèrent, ils ne sont parfois pas capables de traiter les informations eux-mêmes. C'est de ce besoin que sont nées les plates-formes qui vont stocker, agréger et analyser les données à la base du concept de « cloud ».

Le « cloud computing » est un nouveau paradigme de l'informatique. C'est un modèle d'accès à la demande à des ressources, par exemple réseaux, serveurs, stockage, calcul, applications, services, pouvant facilement être configurées en tant que : infrastructure, logiciels et applications [29, 30, 31, 32]. Les applications qui interagissent avec des objets connectés tels que les capteurs offrent un stockage de grandes quantités de données collectées, une énorme puissance de calcul pour un traitement en temps réel de ces données et un réseau à très haut débit pour les rendre accessibles. Les plates-formes s'appuyant sur le cloud permettent aussi aux utilisateurs de se connecter aux objets ambiants de façon simple et transparente, à tout moment et en tout lieu. Le cloud fonctionne finalement comme une interface permettant d'accéder aux services de l'Internet des Objets. Toutefois, ce rapprochement entre les services de l'IoT et ceux du cloud computing pose certains problèmes. En particulier, les problématiques de protection des données personnelles, en terme de confidentialité et d'intégrité et de respect de la vie privée, lors des communications ou du stockage des données, mais aussi les problématiques liées à l'authentification nécessaire des utilisateurs et des services pour assurer les contrôles d'accès adéquats, figurent parmi les défis les plus critiques [33].

1.2.6 L'informatique en proximité (« edge computing »)

Face à la difficulté d'assurer parfois certains de ces défis sécuritaires, et grâce à l'augmentation de la puissance de calcul et de stockage des objets connectés, récemment, le concept de « edge computing » a émergé. Il consiste à ne plus externaliser dans le cloud les traitements et le stockage de données mais à les garder plus proches des objets sur des intermédiaires (des passerelles au sein d'une « smart home » par exemple). Ainsi les données ne sont plus exposées à différentes menaces liées au cloud.

1.3 Les défis de l'IoT

L'augmentation exponentielle du nombre d'objets connectés entraîne parallèlement l'explosion du volume des données générées par ces derniers. Cette évolution engendre de nouveaux défis principalement autour de la sécurisation et l'accessibilité en temps réel aux données. En effet, les enjeux portent sur l'intégration transparente, sécurisée et efficace de l'hétérogénéité des technologies de communications utilisées et des interfaces de connexion associées (les objets connec-

tés doivent pouvoir communiquer facilement non seulement sur le réseau mais aussi entre eux), de l'hétérogénéité de la nature des objets et des technologies qu'ils utilisent, et bien sûr aussi de l'hétérogénéité des données générées selon différents formats. Ainsi, pour permettre son propre développement et accompagner la conception d'applications sûres et efficaces, l'IoT doit à minima relever les défis suivants.

1.3.1 L'hétérogénéité des dispositifs

L'Internet des Objets est composé de dispositifs hétérogènes en termes de ressources, de normes et de standards de communication. En effet, les objets connectés ont souvent des contraintes matérielles (par exemple, puissance de calcul, portée de transmission, capacité de stockage et autonomie énergétique) et logicielles spécifiques selon les applications auxquelles ils sont destinées. Ainsi, une fois déployée, l'objet va récolter des données via différents capteurs et son logiciel embarqué va les traiter. Selon la nature de l'application, le traitement pourra être simple (par exemple, vérifier si la température lue est dans l'intervalle défini préalablement par l'utilisateur) ou complexe (par exemple, détection d'une intrusion dans un bâtiment grâce à l'identification d'un motif particulier dans une vidéo).

Au travers de ces exemples simples, on voit bien l'hétérogénéité des données (simple température versus vidéo) mais aussi la diversité des besoins que cela engendre pour les applications (en terme de puissance de calcul nécessaire, de capacité de stockage, etc.) et bien sûr également la diversité en termes d'exigences de sécurité.

La gestion de l'hétérogénéité des dispositifs est donc essentielle pour garantir l'interopérabilité des différents objets connectés et assurer le fonctionnement efficace et sécurisé de l'IoT.

1.3.2 Les ressources limitées

Historiquement, l'IoT s'est développé au travers de dispositifs intelligents ayant des ressources limitées afin tant de minimiser les coûts que de conserver des objets compacts. Bien sûr aujourd'hui des objets plus puissants sont disponibles grâce aux progrès dans la miniaturisation et l'intégration des composants conjointement à la baisse de leurs coûts de production.

Toutefois, le plus souvent, *objet connecté* rime encore avec *ressources limitées*. Les contraintes de ressources matérielles portent sur la puissance de traitement (CPU et mémoire RAM), la capacité de stockage (mémoire Flash), sur l'autonomie énergétique, sur les communications, etc. L'énergie de fonctionnement est souvent fournie par des piles ou des batteries et parfois, l'autonomie est assurée par des panneaux solaires ou d'autres systèmes de récupération d'énergie. Les communications étant souvent consommatrices d'énergie, les objets supportent souvent une unique technologie de communication telle que celles qui seront présentées dans la section 1.4.

Quant à la sécurité dans l'IoT, on a coutume de dire qu'elle se trouve dans le 'S' de l'acronyme (bien sûr, il n'y en a pas !). Mettre en œuvre la sécurité dans l'IoT doit se faire en prenant en compte ces ressources limitées. En particulier, il faut être attentif à la consommation énergétique (liée aux calculs cryptographiques mais aussi à l'overhead ajouté lors de la transmission des données).

Ne pas prendre en compte ces contraintes conduira inévitablement à l'échec de toute proposition de solution destinée à adresser l'ensemble des IoTs.

1.3.3 La mobilité

De plus en plus fréquemment, dans l'IoT, les dispositifs intelligents sont maintenant capables de se mouvoir. Parmi ces objets, on peut trouver des drones, mais aussi des robots aspirateurs, des véhicules connectés. Afin de fournir un accès permanent et transparent aux données qu'ils produisent mais aussi pour leur permettre d'accéder à d'autres services de l'IoT, ces dispositifs en mouvement (qu'il soit permanent ou ponctuel) conduisent à introduire de la dynamique dans le réseau avec de multiples connexions et déconnexions en temps réel. Il peut s'agir de réseaux dynamiques formés pour un instant par plusieurs dispositifs ou de réseaux statiques (pré-existants) auxquels les dispositifs se rattachent le temps nécessaire.

La gestion de tels réseaux dynamiques devient très difficile dans de grands environnements distribués car déterminer si un dispositif peut être accepté dans le réseau revient à déterminer le niveau de confiance qu'on lui accorde. C'est donc, in fine, une problématique de sécurité.

Dans les réseaux dynamiques, la gestion de la mobilité peut se faire de plusieurs façons. Le dispositif intelligent (dont on rappelle qu'il est intégré à un dispositif physique, c'est-à-dire un objet) doit détecter son mouvement afin d'identifier qu'il va probablement quitter l'emplacement qu'il occupe alors dans la topologie du réseau pour se connecter à un autre endroit ou même à un autre réseau. Cette détection peut se faire par un scan passif des messages des participants aux réseaux ou via le suivi des émissions de balises (« beacons ») [34]. Une autre façon de gérer la mobilité est l'intégration dans les protocoles de messages de signalisation et de contrôle de la localisation des nœuds dans le réseau [35].

Il est à noter que la mobilité augmente la surface d'attaque et qu'elle va par conséquent de paire avec la sécurité. C'est pourquoi une solution de sécurité pour les dispositifs IoT devrait prendre en compte leur mobilité [36].

1.3.4 La sécurité

Les objets intelligents étant souvent déployés dans des environnements ouverts dans lesquels ils ne sont pas surveillés (milieux qu'on considère souvent comme *adverses*). Par exemple, afin de récupérer des données environnementales (température, pression, humidité, luminosité, etc.), si

des capteurs fonctionnant sur batterie sont déployés sur le terrain (pour remonter les données à une station de base), ils ne sont généralement pas sous la surveillance physique à tout instant de l'opérateur les exploitant. En revanche, un adversaire, peut lui y accéder physiquement pour conduire des attaques telles que la copie des données collectées (atteinte en confidentialité), la modification ou la suppression des données collectées (atteinte en intégrité). Les nœuds eux-mêmes peuvent être clonés et/ou modifiés à l'insu de leur propriétaire [36, 37]. Par ailleurs, en raison des ressources limitées des objets de l'IoT, les mécanismes de protections des communications ne sont souvent pas utilisés, ce qui peut permettre à l'adversaire d'écouter les communications, de falsifier des échanges, etc.

Si dans le domaine des applications de surveillance médicale, les dispositifs ubiquitaires comme des pompes à perfusion injectant des médicaments, des caméras de surveillance, des babyphones apportent de plus en plus d'information sur la vie quotidienne des utilisateurs, il convient que ces informations privées le restent. C'est pourquoi de tels dispositifs doivent être sécurisés afin d'empêcher les accès locaux et à distance non autorisés pour protéger les données produites [38].

L'acceptation et l'adoption massive de l'IoT passe obligatoirement par la mise en place des mécanismes de sécurité tant pour protéger les objets connectés, eux-mêmes, que les données qu'ils envoient et reçoivent, car ces mécanismes sont les briques de base pour la conception de solutions IoT sécurisées pour collecter, faire transiter et stocker les informations sensibles [39].

1.4 Les technologies de communication de l'Internet des Objets

Avec l'augmentation du nombre de dispositifs intelligents présents dans la vie quotidienne, le besoin de leur permettre de communiquer, notamment entre eux, est vite apparu. C'est ainsi que pour répondre aux contraintes de ressources (réseau, traitement, stockage, énergie), de nouvelles technologies et de nouveaux protocoles de communication ont été développés ou adaptés. Pour répondre à certains besoins, comme une faible consommation, une portée importante mais aussi une facilité de mise en œuvre, de nouveaux protocoles ont été proposés (par exemple SigFox et LoRaWan qui sont présentés plus loin).

La Figure 1.3 propose une présentation non exhaustive de quelques protocoles et technologies de communication qui peuvent être utilisés pour connecter des dispositifs IoT d'une maison intelligente [28].

Ces technologies et protocoles sont détaillés et comparés ci-après selon les principales couches de la pile réseau auxquelles ils appartiennent.

FIGURE 1.3 – Quelques technologies et protocoles de communication de l'IoT pour une maison intelligente

1.4.1 Les technologies des couches physique et liaison de données

1.4.1.1 X-10

Le protocole filaire X-10 utilise la ligne d'alimentation électrique pour connecter et contrôler différents dispositifs intelligents d'une maison. En effet, il fonctionne sur le principe des CPL (Courants Porteurs en Ligne) pour communiquer entre l'émetteur et le récepteur en envoyant et en recevant des signaux sur les fils d'alimentation électrique [40]. X-10 offre plusieurs avantages : il est simple et facile à implémenter, son installation ne nécessite pas de nouveaux câblages et il permet de relier jusqu'à 256 dispositifs entre eux. Il ne peut en revanche envoyer qu'une seule commande à la fois (sous peine de collision). En ce qui concerne la sécurité du protocole, elle est absente puisque toute personne ayant accès physiquement à la ligne électrique peut envoyer des commandes. Toutefois, il est possible de se prémunir d'attaques extérieures en installant un module de filtrage qui empêchera les signaux émis en amont de l'installation (c'est-à-dire du tableau électrique) de rentrer dans le domicile mais aussi d'éviter que des informations sortent du domicile [4].

1.4.1.2 Bluetooth

Bluetooth est un protocole sans fil, développé par le Bluetooth SIG (« Special Interest Group ») dans le but de simplifier les connexions entre les appareils en supprimant les liaisons filaires. Il offre une connectivité simple et sûre pour une large gamme de dispositifs mobiles et fixes [41]. Les principales caractéristiques du Bluetooth comprennent la robustesse et le faible coût. Cette norme de communication permet l'échange bidirectionnel de données entre différents dispositifs à très courte distance en utilisant des ondes radio sur la bande de fréquences de 2,4 GHz. Bluetooth

4. https ://www.maison-et-domotique.com/5793-le-protocole-x10/

ne cesse de progresser. L'apparition du Bluetooth 4.0 en intégrant un mode basse consommation BLE, (« Bluetooth Low Energy ») a contribué fortement au développement des objets connectés [42]. BLE a pour but de fournir les mêmes fonctionnalités que le Bluetooth classique avec un coût et une consommation d'énergie réduits. Il est ainsi implémenté dans les objets à faible ressource énergétique, et de fait est utilisé pour de multiples applications dans différents domaines tels que la santé. Dans la version 4.1, BLE a introduit l'utilisation de l'algorithme de chiffrement AES (« Advanced Encryption Standard ») avec compteur (CTR, « CounTeR mode ») en mode CBC-MAC (AES-CCM) pour la première fois dans une spécification Bluetooth, afin de fournir à la fois l'authentification et la confidentialité des données transmises. La dernière version de ce standard est Bluetooth 5.0 [43, 44] qui permet de doubler la portée de communication, de multiplier le débit des transmissions basse consommation de 1 à 2 Mbps, d'améliorer l'interopérabilité et de réduire les interférences avec les autres technologies sans fil comme le Wi-Fi.

1.4.1.3 ZigBee

Zigbee est un protocole réseau sans fil spécialement conçu pour les dispositifs limités en ressources [45]. Zigbee se base sur la norme IEEE 802.15.4 et fournit des services de sécurité qui comprennent l'établissement et l'échange de clés, la protection et l'authentification des dispositifs. Il permet également d'utiliser l'algorithme AES avec des clés de 128 bits pour garantir l'intégrité et l'authentification des messages. ZigBee a été largement déployé dans différents domaines comme les IIoT (« Industrial IoT ») et l'agriculture. Mais Zigbee peut aussi permettre aux objets intelligents de collaborer afin d'améliorer la vie quotidienne. Par exemple, dans le domaine médical, Continua Health Alliance [5] utilise le profil standard ZigBee Health Care pour collecter les données à partir de différentes positions (les résidences privées, les centres de loisirs, les maisons de retraite, les hôpitaux, etc.) en raison de son interopérabilité, de sa faible consommation d'énergie mais aussi de sa fiabilité et de sa sécurité.

1.4.1.4 Sigfox

Sigfox est une technologie de communication cellulaire [46] qui offre les mêmes avantages que les réseaux cellulaires téléphoniques (comme la longue portée) tout en permettant aussi aux dispositifs intelligents de ne consommer que peu d'énergie. Sigfox utilise les transmissions UNB (« Ultra Narrow Band ») et chaque transmission n'a besoin que d'une bande passante de 100 Hz. Par défaut, chaque message est transmis à trois reprises, pour augmenter les chances du message d'atteindre la station de base en cas d'absence d'accusé de réception. Par ailleurs, le temps d'attente avant la transmission du message est configurable et chaque message est émis à une fré-

5. http ://www.continuaalliance.org

quence différente des précédents pour diminuer les effets d'affaiblissement du signal [47]. Dans le domaine de la santé, Sigfox peut être utilisé dans les systèmes de détection des chutes et de suivi médical pour surveiller à distance le bien-être du patient. La force de Sigfox réside donc sur la possibilité d'émettre sur de longues distances tout en consommant très peu d'énergie, ce qui allonge la durée de vie de la batterie des objets connectés l'utilisant [48, 49].

1.4.1.5 LoRaWan

LoRaWan (« Long Range Wide-area network ») est un protocole de communication par radio à bas débit et à longue portée. LoRaWan permet à des dispositifs intelligents à faibles ressources de communiquer entre eux en utilisant la technologie LoRa, voire d'être connectés à Internet via des passerelles. L'architecture de base d'un réseau LoRaWan repose sur une topologie en étoile, dans laquelle les passerelles transmettent les messages entre les dispositifs et un serveur central souvent connecté à Internet [50].

Les passerelles sont connectées au serveur du réseau via des connexions standard IP, tandis que les dispositifs sont reliés à une ou plusieurs passerelles via les fréquences radio LoRa et une communication directe (« one hop »). Les communications sur ce réseau sont généralement bidirectionnelles, sachant que la communication à partir des dispositifs vers le serveur du réseau constitue généralement l'essentiel du trafic. Les dispositifs communiquent avec les passerelles en utilisant les différents canaux de fréquences et les débits de données disponibles. Pour maximiser la durée de vie des batteries des dispositifs et la capacité globale du réseau, l'infrastructure réseau LoRa utilise un schéma de débit de données adaptatif, ADR (« Adaptive Data Rate »).

Dans le domaine de la santé, les dispositifs utilisant LoRaWAN peuvent être utilisés pour surveiller la pression artérielle et le taux de glucose des patients à l'hôpital ou à leur domicile, pour surveiller les personnes âgées à domicile afin de détecter des chutes ou des pertes de connaissance [51, 52, 53].

1.4.1.6 Z-Wave

Z-Wave est un protocole de communication sans fil principalement destiné à la domotique qui permet aux dispositifs d'être connectés à un réseau de type mesh via des communications radio [54]. Il offre une transmission fiable à partir du contrôleur du réseau (une unité de commande) vers un ou plusieurs dispositifs.

Chaque réseau Z-Wave ayant son propre identifiant, plusieurs réseaux Z-Wave peuvent fonctionner dans un même endroit sans interférence. Les dispositifs d'un même réseau peuvent être utilisés soit de manière autonome et décentralisée ou bien de manière centralisée à l'aide d'un contrôleur central ou d'une passerelle IP intégrée afin d'ouvrir l'accès au réseau Z-Wave à d'autres

réseaux (Internet par exemple). Z-Wave est utilisé dans les maisons intelligentes pour différentes applications, telles que le contrôle de la lumière, des appareils électroménagers, de l'énergie et de plus en plus souvent pour le contrôle des dispositifs médicaux portés [55, 56].

1.4.1.7 Wi-Fi

Le Wi-Fi (« Wireless-Fidelity ») désigne une technologie de communication sans fil normalisée par le groupe IEEE 802.11. Il est principalement utilisé pour relier des appareils informatiques à des réseaux locaux WLAN (« Wireless Local Area Network») permettant la transmission de données haut débit [7]. Il remplace ainsi la communication via des supports filaires classiques tels que le câble Ethernet par des ondes radio.

Le Wi-Fi est devenu omniprésent dans notre quotidien. Il permet aux utilisateurs d'accéder à Internet tout en conservant leur mobilité grâce à l'utilisation de différents dispositifs comme le smartphone et les PCs. Cependant, comme initialement il a été optimisé pour le transfert de données volumineuses et non pour une faible consommation énergétique, cette dernière a longtemps été une contrainte majeure à son adoption pour des dispositifs à faible puissance.

Ces dernières années, de nouveaux standards sont apparus tels que les normes IEEE 802.11n et IEEE 802.11ac ou la norme IEEE 8012.11ah utilisant des fréquences plus basses avec une portée plus longue et intégrant des fonctionnalités d'économie d'énergie [57]. L'utilisation du Wi-Fi par les dispositifs IoT offre une connexion directe à Internet réduisant la complexité de l'architecture des systèmes IoT. Notamment, dans une maison intelligente on trouve plusieurs dispositifs IoT pouvant utiliser le Wi-Fi et que l'on peut classer selon leur source d'énergie en différentes catégories [58, 59] :

- Dispositifs alimentés par le réseau électrique : ce sont des dispositifs qui n'ont pas de contraintes d'énergie (comme les appareils électroménagers, les PCs) .
- Dispositifs rechargeables : ce sont des dispositifs qui fonctionnent avec des batteries rechargeables dont l'autonomie peut durer jusqu'à quelques jours (comme le smartphone, la tablette).
- Dispositifs alimentés par piles : ce sont des dispositifs à faible consommation qui disposent de batteries standards (comme les capteurs : les détecteurs de fumée, les détecteurs de mouvement et les capteurs de température) qui doivent fonctionner pendant une longue période.

1.4.1.8 Comparaison entre les différentes technologies

Le Tableau 1.1 résume les principales différences entre les différentes technologies de communication sans fil présentées.

TABLEAU 1.1 – Comparaison entre différentes technologies de communication sans fil utilisées dans l'IoT

Protocole	**Bluetooth**	**ZigBee**	**Z-Wave**	**LoRaWAN**	**Sigfox**	**Wi-Fi (802.11ac)**
Débits	2 Mbps	250 kbps	100 kbps	0,3 - 50 kbps	1 Mbps	433 Mbps - 1300 Mbps
Sécurité	AES 128 bits	AES 128 bits	AES 128 bits	AES 128 bits	Partiellement adressée	WEP - WPA (AES 128 bits)
Fréquences	2,4 GHz	2,4 GHz	868 MHz (EU) 908 MHz (USA)	868 MHz (EU) 915 MHz (USA)	868 MHz (EU) 902 MHz (USA)	5 GHz
Portée	10 - 100 m	10 - 100 m	30 m (à l'intérieur) et 100 m (à l'extérieur)	20 km (zone rurale) et 8 km (zone urbaine)	50 km (zone rurale) et 10 km (zone urbaine)	35 m (à l'intérieur) et 300 m (à l'extérieur)
Topologies	Réseau en étoile ou bus	Réseau en étoile ou mesh	Réseau mesh	Réseau en étoile	Réseau en étoile	Infrastructure ou Ad hoc

1.4.2 Les technologies de la couche réseau

Dans cette couche, nous présentons seulement le protocole 6LoWPAN.

6LoWPAN est un protocole développé par un groupe de l'IETF (« Internet Engineering Task Force ») et destiné aux dispositifs à faibles ressources [60]. Ce protocole permet l'utilisation efficace d'IPv6 sur des réseaux sans fil bas débit de courte portée utilisés par des dispositifs contraints.

En raison de la taille par défaut du MTU (« Maximum Transmission Unit ») IPv6 de 1280 octets, un paquet IPv6 serait trop grand pour tenir dans une trame IEEE 802.15.4 sans qu'il soit fragmenté. Pour cela, le protocole 6LoWPAN offre la possibilité de transmettre des paquets IPv6 sur IEEE 802.15.4 en mettant en œuvre des mécanismes de fragmentation et de réassemblage des paquets.

Le routage sur des environnements 6LoWPAN est pris en charge par le protocole RPL (« Routing Protocol for Low power and Lossy Networks ») [61]. RPL a été conçu spécialement pour les réseaux de nœuds disposant de peu de ressources. Ce protocole permet de créer rapidement des itinéraires réseau, de partager les connaissances en matière de routage et d'adapter la topologie de manière efficace [62]. Ainsi, il satisfait les besoins de certains domaines d'application de l'IoT comme l'automatisation des bâtiments et maisons connectés (chauffage, ventilation et climati-

sation, éclairage, contrôle d'accès, protection incendie) mais aussi la surveillance de l'environnement (réseaux de capteurs). RPL peut aussi être utilisé dans les applications médicales car il offre un temps de réponse optimal et des itinéraires de routage sont disponibles à la demande [63, 64]. Ainsi, dans les maisons intelligentes, 6LoWPAN permet de connecter des dispositifs contraints mesurant les données physiologiques de patient à un réseau externe (IP) afin que les médecins et/ou les personnes concernées puissent surveiller en temps réel l'état du patient [65, 66, 67].

1.4.3 Les technologies de la couche transport

1.4.3.1 MQTT

Message Queuing Telemetry Transport [68] est un protocole de communication conçu pour les communications M2M (« Machine-to-Machine ») par IBM (« International Business Machines »). MQTT est ouvert, simple, léger et facile à mettre en œuvre. Il consomme peu d'énergie et ne nécessite que peu de ressources processeur et mémoire. Ces caractéristiques le rendent idéal pour une utilisation dans des environnements contraints. Le protocole MQTT peut par exemple être utilisé pour le contrôle des systèmes domotiques d'une maison intelligente. Dans [69], les auteurs automatisent le comportement des objets du quotidien (capteurs, volet, frigos, vidéo surveillance) pour qu'ils fournissent des services ou communiquent des informations sur leur état et/ou leur environnement. MQTT fonctionne selon le principe du modèle client/serveur où chaque dispositif (client) établit une connexion TCP à un serveur. Il s'appuie sur le modèle producteur/consommateur et plus précisément « publish/subscribe ». Le serveur, appelé « broker » agit dans ce modèle comme un intermédiaire de mise en relation entre « publisher » et « subscriber ». Un client peut donc être soit l'objet qui communique l'information au broker (il est alors appelé « publisher ») ou bien celui qui demande l'information au broker (il est alors appelé « subscriber »). MQTT est orienté message. Les messages sont publiés sur des « topics » (qui sont des sortes de chemins d'accès à des ressources) hébergés par le broker. Les « subscribers » peuvent alors souscrire auprès du broker à plusieurs topics et ils seront ainsi notifiés de tous les messages reçus par le broker pour les topics pertinents. Par exemple, sur un topic nommé `température`, plusieurs clients (« publishers ») publient leurs données de température en intercalant leur identifiant client sur leur topic, un autre client « subscriber » peut alors souscrire auprès du broker au topic `température` et ainsi, il recevra les températures des clients ayant publié. Dans MQTT, les échanges de données peuvent être sécurisés, soit en utilisant SSL (« Secure Socket Layer ») ou TLS (« Transport Layer Security ») et l'authentification se fait soit par certificats SSL/TLS soit par login/mot de passe. Idéalement, il y a une authentification mutuelle entre le client (« publisher » ou « subscriber ») et le broker afin d'assurer la confidentialité et l'intégrité des données échangées. Un intérêt supplémentaire de MQTT est de permettre au « publisher » de définir la qualité de son message via la notion de

QoS (« Quality-of-Service ») selon 3 niveaux [70].

1.4.3.2 CoAP

Constrained Application Protocol est un protocole de communication élaboré par le groupe CoRE (« Constrained Resource Environments ») de l'IETF (« Internet Engineering Task Force ») pour étendre l'architecture web aux applications M2M [71]. Il est basé sur UDP pour minimiser la surcharge réseau et consomme peu d'énergie. Ce protocole a été développé pour assurer la fiabilité des communications dans des environnements à faible bande passante entre des capteurs ou des actionneurs. Dans [72] une utilisation dans le cadre des maisons intelligentes est présentée. CoAP peut être vu comme un sous ensemble de HTTP (« HyperText Transfer Protocol »). Il suit le modèle client/serveur où les clients utilisent les méthodes `GET`, `PUT`, `POST` et `DELETE` pour accéder aux ressources des serveurs. Il fonctionne de manière asynchrone et les messages échangés sur le réseau sont soit acquittés par le récepteur, soit envoyés sans garantie de réception. Pour sécuriser les transferts de données sur UDP, le protocole CoAP utilise DTLS (« Datagram Transport Layer Security ») pour offrir les mêmes garanties que TLS pour le protocole TCP. Il est à noter que DTLS adresse aussi les problèmes de perte et de réordonnancement des paquets.

1.4.3.3 Comparaison entre MQTT et CoAP

Les principales différences entre CoAP et MQTT sont résumées dans le Tableau 1.2. La première différence entre CoAP et MQTT est que le premier utilise le protocole UDP, tandis que le second fonctionne sur TCP. Comme UDP n'offre pas de garantie de fiabilité lors de la communication, CoAP fournit son propre mécanisme de fiabilité en utilisant 4 types de messages. Les messages dit *confirmables* nécessitent un accusé de réception tandis que les messages dit *non-confirmable* n'en ont pas besoin [73]. Les autres messages sont les *accusés de réception* et ceux de *reset*. Au niveau de la sécurité, MQTT utilise le protocole TLS (« Transport Layer Security ») alors que CoAP implémente le protocole DTLS.

1.4.4 Les technologies de la couche application

Pour la couche application, nous nous intéresserons seulement à Thread [6], un protocole de réseau mesh sans fil, conçu principalement dans le but de connecter en toute sécurité et de manière fiable les dispositifs intelligents d'une maison [74]. Thread peut gérer plus de 250 nœuds appartenant à un même réseau. Il se caractérise par une faible latence des communications (moins de 100 millisecondes). Thread peut supporter plusieurs couches applicatives qui s'exécutent sur IPv6 comme CoAP et MQTT [75].

6. https ://www.infoq.com/articles/thread-protocol-for-home-automation

TABLEAU 1.2 – Comparaison entre les protocoles MQTT et COaP

Protocole	**MQTT**	**CoAP**
Couche transport	TCP	UDP
Fiabilité/QoS	3 niveaux de qualité de services	4 types de messages : confirmable, non-confirmable, AcK et Rst
Architecture	Communication many-to-many entre les clients via le broker	Communication one-to-one entre le client et le serveur
Sécurité	Utilise TLS	Utilise DTLS

La pile de protocoles Thread est un standard ouvert pour des communications M2M fiables, sans fil et de faibles puissances. Elle se caractérise principalement par l'utilisation de protocoles simples pour constituer, joindre, et maintenir des réseaux Thread. Ces protocoles permettent aussi aux systèmes de s'autoconfigurer et de résoudre les problèmes de routage quand ils se produisent, comme dans le cas d'échec ou de perte de dispositifs individuels. Les différents dispositifs communiquent sur le réseau Thread via le protocole UDP. L'implémentation du protocole DTLS permet d'authentifier les dispositifs et de sécuriser les échanges de messages sur le réseau.

1.5 Deux technologies clés de l'IoT

Les systèmes WSN et RFID sont deux technologies clés de l'IoT. Leur intégration aux objets intelligents offrent de nouvelles capacités de communication. Ces dernières années, de par leurs avantages considérables et leur utilisation dans l'IoT, ces technologies ont été de plus en plus utilisées et nous les présentons donc en détail dans cette section.

1.5.1 La technologie RFID

Dans le contexte de l'Internet des Objets, la technologie RFID (« Radio Frequency IDentification ») permet, via les fréquences radio, l'identification automatique à distance d'objets équipés d'étiquettes RFID. En effet, un système RFID est principalement composé de deux entités qui communiquent entre elles, le lecteur (aussi appelé interrogateur) et l'étiquette (aussi appelée « tag » en anglais). Dans sa version passive, le lecteur va interroger le tag via des ondes radio et celui-ci utilisera alors l'énergie électromagnétique transportée par le signal reçu pour s'alimenter et, à l'issue du traitement, envoyer ses données en réponse au lecteur [76].

- Les tags RFID : Ceux sont de petits circuits intégrés apposés sur l'élément à identifier. Ils peuvent communiquer avec des lecteurs sur différentes fréquences selon leurs domaines

d'application. Les systèmes RFID basse fréquence (BF, 125-134 kHz) et haute fréquence (HF, 13,56 MHz) sont des systèmes à courte portée qui reposent sur le couplage inductif entre l'antenne (c'est-à-dire une bobine) du lecteur et l'antenne de l'étiquette via un champ magnétique. Les systèmes RFID à ultra-haute fréquence (UHF, 860-960 MHz) et à micro-ondes (2,4 GHz et 5,8 GHz) sont des systèmes d'interrogation longue portée qui utilisent des ondes électromagnétiques se propageant entre le lecteur et l'étiquette [77]. Un tag RFID peut être en lecture seule ou réinscriptible (lecture et écriture avec mémoire).

Si ici nous n'avons décrit que le fonctionnement de la technologie RFID passive, il existe d'autres types de RFID, et en particulier il est possible de trouver une description complète de la technologie RFID active dans le mémoire de thèse d'Amina Cherif [78]. En substance, contrairement au tag RFID passif qui reçoit son énergie du signal provenant du lecteur et est donc seulement esclave dans la communication, les tags actifs sont équipés d'une source d'alimentation (par exemple une batterie) et ils peuvent par conséquent initier la communication.

- Les lecteurs RFID : Parfois aussi nommés interrogateurs, en raison de leur action d'interrogation des tags, les lecteurs sont souvent des intermédiaires vers une base de données hébergée sur un serveur local ou sur le cloud.

L'identification par radio-fréquences reste une technologie prometteuse pour la mise en œuvre des systèmes de santé intelligents. En outre, en plus d'être une technologie fondatrice de l'IoT, c'est un moyen efficace d'identifier et de gérer les objets de manière unique. L'utilisation de la

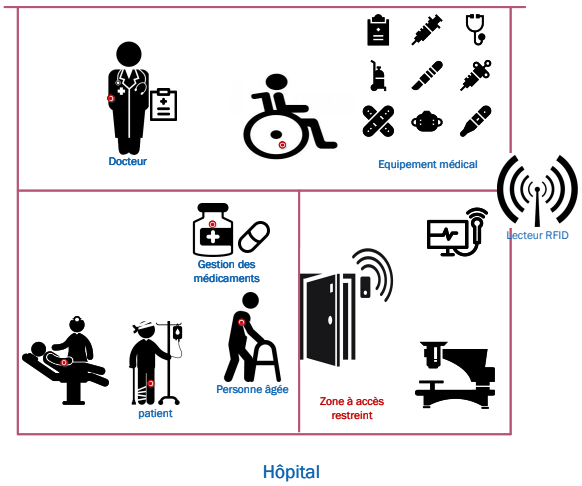


FIGURE 1.4 – Technologie RFID appliquée dans un milieu hospitalier

technologie RFID dans le secteur de la santé [79, 80, 81] permet aux fournisseurs de services de santé de faciliter la prise de décisions et l'accomplissement de tâches habituellement difficiles dans un environnement clinique complexe. Les applications RFID hospitalières sont multiples

comme le montre la Figure 1.4.

- Suivi et gestion des équipements médicaux.
- Identification et suivi du personnel médical (médecins, infirmières, etc.).
- Identification et surveillance des patients et de leurs mouvements.
- Contrôle d'accès par l'identification des infirmières et des médecins entrant dans les salles d'opération, salles de médecine, etc.
- Suivi de la stérilisation et surveillance des outils chirurgicaux.

1.5.2 Les réseaux de capteurs

Un réseau de capteurs WSN (« Wireless Sensor Network ») est constitué d'un ensemble de nœuds capteurs auto-alimentés dotés de capacités de calcul et de communication sans fil. Les capteurs autonomes disposent souvent de ressources énergétiques limitées et sont capables de collecter, traiter, analyser et disséminer des informations via des ondes radio (par exemple via la technologie ZigBee). Un nœud capteur est composé généralement d'interfaces de capture de l'information (c'est-à-dire de mesure d'une grandeur), d'un microprocesseur, d'une unité mémoire, d'une interface de communication et d'une batterie comme source d'énergie. Ces capteurs peuvent être attachés à des objets/personnes ou déployés dans l'environnement selon les besoins applicatifs. Il existe plusieurs types de capteurs selon les phénomènes surveillés ou le type de données récoltées comme les capteurs de température, d'humidité, de position ou de lumière.

Dans un environnement médical [82, 83, 84], les WSNs sont devenus indispensables, y compris dans les maisons intelligentes (comme illustré par la Figure 1.2) car ils peuvent être utilisés pour mesurer des paramètres physiques de la personne surveillée (c'est-à-dire un patient) tels que les battements du cœur, la pression artérielle, la température corporelle, les signaux cérébraux. Les informations collectées et remontées via le réseau à la station de base sont ensuite traitées et analysées par le système avant d'être transférées par la station de base via Internet aux différentes acteurs (médecin, services de secours, etc.). Ce type de réseaux permet donc aux professionnels de santé de surveiller à distance le patient sur la base des informations reçues.

1.5.2.1 Les réseaux de capteurs corporels

Afin d'améliorer le suivi des constantes physiologiques du patient, les systèmes de santé utilisent de plus en plus l'IoT au travers de la technologie des réseaux de capteurs corporels, BSN (« Body Sensor Network »[7]) [85]. Il s'agit essentiellement d'un ensemble de nœuds capteurs sans

7. Dans la littérature le terme de BAN (« Body Area Network ») est utilisé de manière équivalente ; de la même façon, que les termes WBAN (« Wireless Body Area Network »), WBASN (« Wireless Body Area Sensor Network »), BASN (« Body Area Sensor Network »). Pour notre part, nous utiliserons le terme BSN.

fil légers et à faible consommation d'énergie utilisés pour surveiller les fonctions du corps humain et son environnement. Comme illustré Figure 1.5, les nœuds du BSN sont utilisés pour collecter différentes informations biologiques comme la fréquence cardiage d'un électrocardiogramme (ECG), la pression artérielle, la fréquence respiratoire, la saturation en oxygène (par exemple avec l'oxymétrie de pouls), la température.

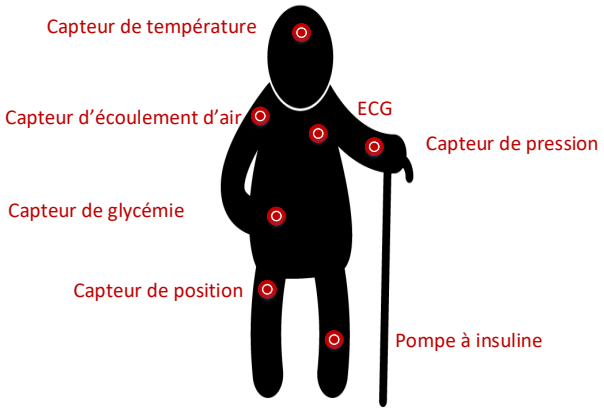


FIGURE 1.5 – Exemple d'un réseau de capteurs corporels

Les récents progrès des technologies sans fil ont permis de passer des systèmes BSN conventionnels câblés, utilisés principalement dans les hôpitaux et qui sont contraignants pour les patients lors de leurs déplacements, à des systèmes sans fil offrant plus de mobilité et de moins contraintes dans les mouvements, permettant ainsi d'avoir un suivi continu lors des différentes activités quotidiennes.

Selon les technologies de communication utilisées, les différentes données corporelles du patient provenant des capteurs disposés autour ou dans le corps humain, sont soit transmises à une station de base, située à quelques dizaines de mètres seulement, qui les relaie à un centre de diagnostic distant, soit elles sont acheminées directement dans le « cloud » (c'est-à-dire sur un centre de données, « data center » en anglais, distant) où tourne une application permettant le diagnostic par les soignants. L'objectif de la transmission de ces données est la prise de décision de l'équipe médicale : par exemple déclencher une action sur une pompe à insuline connectée portée par le patient.

1.6 L'IoT dans le domaine médical

L'appropriation des technologies de l'Internet des Objets dans la vie quotidienne du grand public à travers la domotique et les maisons intelligentes (Figure 1.2) ainsi que le phénomène d'automesure (« quantified-self », Figure 1.1) offre de nouvelles perspectives pour l'amélioration de la

qualité de vie des utilisateurs. En effet, avec l'augmentation de l'espérance de vie des personnes âgées, les maisons intelligentes permettent de répondre à leur désir de poursuivre leur fin de vie à domicile aussi longtemps que possible, plutôt que dans un établissement spécialisé.

Le « quantified-self » consiste à quantifier le corps et ses activités grâce à des capteurs placés (portés ou incorporés) sur le corps. Précisément, il s'agit de capturer, d'analyser et de stocker ces données corporelles (son métabolisme, ses comportements, ses mouvements, ses consommations) en vue d'améliorer son bien-être ou de contrôler sa santé. De nombreuses applications ont vu le jour dans le domaine de la santé comme permettre de surveiller la tension artérielle, le rythme cardiaque, le niveau d'activité physique et la glycémie.

L'intégration des technologies de l'IoT dans le domaine médical a contribué à l'évolution de la santé. Historiquement centralisés et principalement présents dans les hôpitaux, ces systèmes sont devenus plus ubiquitaires et intelligents et ont été distribués dans différentes localisations comme les maisons intelligentes. Dans le secteur de la santé, récemment une transition des services s'est opérée passant de systèmes reposant purement sur le diagnostique humain vers des solutions reposant sur la technologie [86, 87]. C'est ainsi que sont apparues des solutions novatrices et efficaces pour le suivi et le traitement à domicile des personnes âgées, des personnes à activités réduites et des patients atteints de maladies chroniques [88]. Ces solutions reposent sur divers dispositifs intelligents qui récoltent et transmettent via le réseau les paramètres de santé des personnes surveillées (par exemple le rythme cardiaque ou la pression sanguine) aux personnes concernées (médecin traitant, membres de la famille, etc.). Les données recueillies à partir de ces dispositifs peuvent aussi être stockées et analysées par des médecins spécialistes pour élaborer leurs diagnostics concernant l'état de santé des patients surveillés (quelque soit le lieu où ils sont) et pour réagir plus rapidement en cas d'alerte. L'utilisation de lecteurs personnels de glycémie connectés via des services dans le cloud permettent aux personnes diabétiques de connaître précisément leur glycémie, de contrôler leurs injections d'insuline et de leur fournir des informations et des analyses complémentaires. L'IoT permet aussi d'assister une personne âgée dans des tâches quotidiennes, via un environnement intelligent, par exemple en allumant et éteignant les lumières ou le chauffage. L'IoT peut donc aider une personne âgée à demeurer chez elle plutôt que de devoir partir vivre en maison de retraite [89]. En résumé, l'IoT offre toutes les caractéristiques et les fonctionnalités nécessaires pour améliorer le domaine de la santé et il contribue à la réalisation de la vision de l'assistance à la vie ambiante, AAD.

1.7 La santé et l'Assistance à l'Autonomie à Domicile

De nos jours, de plus en plus de personnes âgées tentent de vivre de manière autonome dans leur propre maison. Cependant, avec l'âge, rester en sécurité et indépendant est un défi majeur.

L'assistance à la vie ambiante a parmi ses principaux objectifs d'aider ces personnes âgées à atteindre cette autonomie. Elle vise aussi à assister les personnes à activités réduites (handicapées – par exemple les personnes en fauteuil roulant) ou, en général, les personnes qui ont simplement besoin d'une aide supplémentaire pour vivre seules. Les systèmes d'AAD peuvent aider les personnes à ne pas oublier leurs médicaments, à surveiller leurs modes de vie et leur santé, mais également à faire en sorte qu'elles se sentent moins isolées en proposant des services sociaux et des divertissements. L'IoT se présente donc comme une solution intéressante pour y parvenir [90, 91, 92].

1.7.1 L'assistance à l'autonomie dans les maisons intelligentes

L'assistance à la vie ambiante appliquée dans les maisons intelligentes bénéficie de différentes technologies de l'IoT, comme les capteurs ambiants, la reconnaissance visuelle, les systèmes de sécurité (détection de chute, vidéo surveillance) ce qui permet la surveillance de l'état de santé et des activités des personnes à travers plusieurs services comme l'illustre la Figure 1.6.

FIGURE 1.6 – Services de l'assistance à l'autonomie dans un domicile intelligent

1. **Détection**

   La surveillance continue de la personne par des capteurs ambiants, portés ou incorporés aide à détecter les anomalies et les changements soudains de ses données corporelles et à réagir selon le degré de l'alerte constatée. Par exemple, dans un scénario où la personne enregistre ses prises de médicaments à un moment précis de la journée quotidiennement, un oubli lui sera automatiquement signalé. Un second exemple, la détection de la dégradation soudaine de l'état de santé en cas de crise cardiaque va permettre d'alerter les services d'intervention d'urgence en temps réel [93, 94].

2. **Inclusion**

   L'AAD permet de s'assurer que la personne dans ses activités quotidiennes a une activité physique via la détection de sa position et des mouvements qu'elle effectue. Grâce aux technologies de l'IoT, il est ainsi possible de savoir si la personne est en train de faire du sport, de dormir ou de regarder la télévision, afin de s'assurer de son bien-être. Ceci permet également l'inclusion sociale des personnes handicapées, par exemple en leur offrant un support et un accès aux activités qu'elles préfèrent et souhaitent. En effet, l'utilisation de technologies de l'IoT permet donc aux personnes âgées et/ou handicapées de : (1) dépasser leur handicap sans être marginalisées par la société ; (2) changer la perception de la société selon laquelle elles seraient limitées dans leurs activités quotidiennes selon le niveau d'aide que leur apporte leur famille ; (3) rester actives et de vivre de manière indépendante le plus longtemps possible [95, 96, 97].

3. **Prévention**

   En termes de prévention, les données corporelles et environnementales récoltées à partir de la surveillance du métabolisme de la personne par le biais de ses signes vitaux, des vidéos récoltées, ainsi que les données de son environnement peuvent être interprétées avec les données existantes du patient. Avec toutes ces informations, l'analyse de son état de santé permet de considérablement améliorer le diagnostic, le traitement, la gestion et la guérison de différentes maladies. C'est pourquoi les systèmes AAD commencent à être de plus en plus déployés pour différentes situations telles que les chutes, l'immobilité physique, la surveillance des activités de la vie quotidienne, l'occupation des espaces domestiques, l'analyse du comportement. L'AAD est donc une solution très intéressante pour permettre de prévenir les risques sur la santé, pour détecter de façon précoce des maladies potentielles qui pourraient se développer, même si évidemment son efficacité dépend aussi de l'analyse et du traitement des données récoltées [98, 93].

4. **Support**

   Les systèmes de l'AAD sont destinés à aider les personnes âgées et/ou handicapées à pro-

mouvoir leur autonomie, leur indépendance, leur sécurité, leur participation sociale et leur bien-être. Ainsi, l'AAD permet de suivre les patients à leur domicile au lieu de les hospitaliser, comme par exemple en cas de réhabilitation physique après un accident, tant sur l'aspect purement physique (on pourrait dire mécanique) que sur l'état émotionnel du patient via la détection émotionnelle afin de l'aider aussi psychologiquement dans un contexte de soins de santé à domicile. En effet, identifier et comprendre le comportement émotionnel des personnes, en situation de réhabilitation et/ou d'handicap permet d'optimiser la relation d'aide et le suivi psychologique qui sont cruciaux pour pouvoir se remettre d'une maladie. L'AAD a donc le potentiel d'offrir un soutien aux patients, à toutes les étapes de l'évolution de leur santé, de la prévention au suivi en passant par le traitement, la rééducation et la gestion du mode de vie en général et du bien-être [99, 100, 101, 102].

5. **Supervision**

   Les systèmes de l'assistance à la vie ambiante comprennent des dispositifs de surveillance médicale permettant de contrôler et de surveiller des conditions médicales spécifiques comme par exemple l'utilisation de la télésurveillance à domicile pour les patients atteints de maladies chroniques, la surveillance de la composition sanguine et l'administration de médicaments pour les patients diabétiques, la surveillance multi-paramétrique des personnes atteintes de maladies cardiovasculaires, pulmonaires, d'asthme et de troubles du sommeil, la détection et la surveillance des performances des activités pour les maladies neuromotrices. En ce qui concerne la sécurité et la protection de l'environnement du domicile, les dispositifs de détection tels que les capteurs de mouvement, de température, de fumée détectent les conditions anormales et informent les personnes concernées en vue de permettre une intervention efficace [103, 104, 105, 106].

1.8 Un modèle d'architecture IoT pour les applications de santé et d'AAD

Afin de présenter un modèle d'architecture IoT pour les applications de santé et d'AAD, nous illustrons par la Figure 1.7 le scénario d'une maison intelligente car, par définition, les soins de santé d'un tel scénario concernent différents domaines, comme la prévention des maladies, le maintien et l'amélioration de l'état de santé, ainsi que la garantie du bien-être des personnes. Dans le cadre de l'AAD, comme nous l'avons vu dans la section 1.2.4 les maisons intelligentes offrent plus d'indépendance à leurs utilisateurs et améliorent leur qualité de vie. En effet, elles possèdent en général un réseau local HAN (« Home Area Network ») hébergeant divers objets intelligents (lampe intelligente, horloge intelligente, capteurs de température, etc.) qui peuvent communiquer directement de différentes manières, sans intervention humaine afin d'offrir différents services.

Dans ce type de modèle d'architecture IoT, on retrouve les deux principaux modes de communication entre objets connectés que l'on avait cités dans l'introduction de ce chapitre : à savoir le Machine à Machine M2M (« Machine-to-Machine ») et le Machine à Cloud M2C (« Machine-to-Cloud »).

De façon générale, le type de communication M2M permet aux objets intelligents d'échanger et de partager des données de manière décentralisée indépendamment d'un système centralisé [13, 14, 15]. Dans une maison intelligente, c'est par exemple un capteur de lumière détectant la baisse de la luminosité et envoyant alors un message pour allumer une lampe intelligente sans aucune intervention humaine.

À l'inverse le type de communication M2C est centralisé entre les objets intelligents et le cloud. Dans le cadre d'une maison intelligente, il peut s'agir d'un thermostat intelligent capable de se connecter au service de prévisions météorologiques disponible sur le cloud pour déterminer les actions à effectuer afin de conserver la température demandée par l'utilisateur (c'est-à-dire s'il doit par la suite envoyer, ou pas, un ordre M2M à l'appareil concerné (chauffage, climatisation, etc.)).

Ainsi, si l'on se concentre sur l'exemple des soins de santé et d'AAD présentés Figure 1.7, plusieurs objets intelligents sont présents dans la maison intelligente : a) les appareils médicaux portés par le patient, b) les appareils environnementaux (capteurs et actionneurs) et c) les appareils ménagers et multimédias connectés. Comme nous venons de le voir, ces objets intelligents collaborent et communiquent de différentes manières. Par exemple, les dispositifs portables sur le corps humain forment un réseau BAN (« Body Area Network ») et sont généralement connectés via le mode de communication M2M à un point d'accès unique pouvant être un point d'accès externe tel que la passerelle domestique ou une unité centrale BCU (« Body Central Unit »), un dispositif porté par le patient pour collecter les informations corporelles du patient auprès des divers capteurs corporels. Ensuite, les données sont soit transmises à un dispositif de gestion local tel que le téléphone portable (smartphone) du patient, soit directement à des serveurs distants dans le cloud. Le mode de communication M2C est alors utilisé pour permettre aux médecins agréés de surveiller l'état de santé du patient et de récupérer ses données médicales au besoin.

La sécurisation de ces deux modes de communication, M2M et M2C, sera au cœur de nos travaux des chapitres 3 à 5.

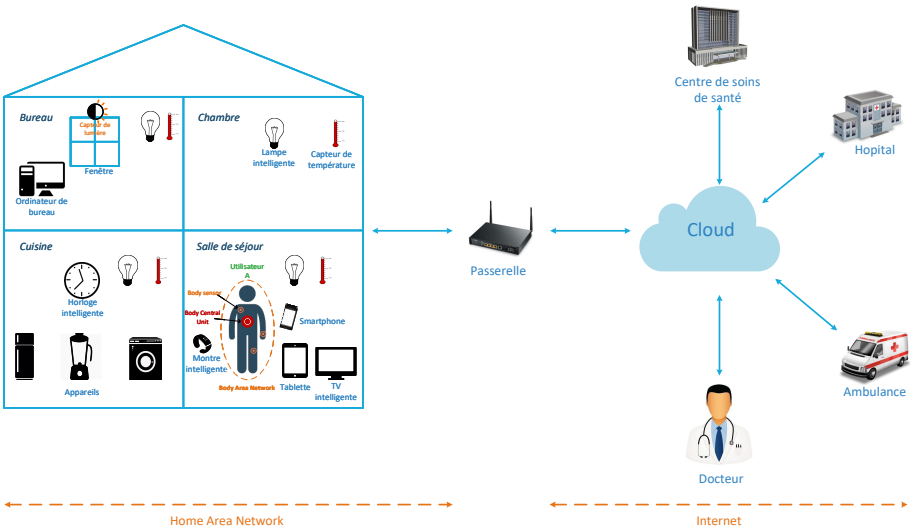


FIGURE 1.7 – Applications de santé et d'AAD dans un scénario d'une maison intelligente

1.9 Conclusion

Dans ce premier chapitre nous avons présenté l'Internet des Objets en général et les différentes notions auxquelles nous nous intéressons particulièrement au sein de cette thèse. Par la suite, nous avons introduit l'application des technologies de l'IoT au domaine médical en général et dans les hôpitaux et les maisons intelligentes en particulier via l'assistance à la vie ambiante tout en montrant les bénéfices qu'elle peut apporter pour l'amélioration de la vie quotidienne des personnes âgées et/ou handicapés et/ou ayant des activités réduites. En effet, l'adoption importante des objets intelligents comme les capteurs biomédicaux permet une surveillance constante de ces personnes dépendantes facilitant ainsi leur prise en charge par les services médicaux. C'est dans ce contexte d'applications de santé et d'AAD que nous avons introduit les deux modes de communication M2M et M2C. En effet, ces dispositifs collectant et manipulant des informations personnelles hautement sensibles (comportement de la personne, ses données de santé, etc.), il est nécessaire d'assurer un haut niveau de sécurité afin que les données ne soient accessibles que pour les personnes autorisées.

Ainsi, dans le prochain chapitre, nous allons nous intéresser aux menaces, aux propriétés de sécurité et aux mécanismes de sécurité dans l'Internet des Objets mais aussi au respect de la vie privée des utilisateurs dans le domaine médical.

Chapitre 2

La sécurité et le respect de la vie privée dans le domaine médical

Sommaire

2.1 Introduction

Avec le développement de l'Internet des Objets dans le domaine médical, les dispositifs connectés se retrouvent dans de nombreuses applications de e-santé (e-health) pour permettre, par exemple, le suivi médical régulier des patients. À l'évidence, ces dispositifs intelligents, sans fil et à bas coût de l'IoT sont des atouts indéniables pour le domaine de la santé puisque l'IoT permet d'interconnecter les objets et les personnes afin de faciliter la circulation de l'information et permettre que celle-ci soit accessible de n'importe où et à n'importe quel moment. Ainsi, ces objets connectés, que ce soit un pilulier, un tensiomètre, une balance, un oxymètre, un glucomètre, etc., sont amenés à traiter des données personnelles des patients puis à les envoyer via une liaison sans fil (par exemple : BLE) vers, par exemple, une application dédiée installée sur le smartphone ou la tablette de l'utilisateur. L'utilisateur peut ainsi prendre connaissance de ces informations (mesures, paramètres physiologiques, suivi de la posologie, etc.) en temps réel et peut en plus consulter son historique médical. L'utilisation de ces technologies pour le suivi quotidien d'un patient permet également de mieux préparer son hospitalisation (c'est une phase de pré-hospitalisation). Ces technologies peuvent aussi offrir d'autres services comme la confirmation des rendez-vous, l'échanges de documents, le suivi de l'usage d'un dispositif connecté, le suivi post-hospitalisation, etc. permettant ainsi de renforcer le lien entre le corps médical et les patients.
De façon générale, l'IoT permet une meilleure prise en charge médicale du patient [107, 108]. En effet, les dispositifs intelligents récoltent et communiquent des informations pertinentes sur l'état de santé de la personne au quotidien, ce qui constituent des données d'analyse qui concourent à aider au suivi continu du patient, à la fiabilisation des diagnostics médicaux, et permet aussi d'améliorer le suivi des traitements médicaux.
Néanmoins, malgré les nombreux avantages qu'ils apportent, les objets de l'IoT présentent également un certain nombre de menaces qui ralentissent leurs essors ; menaces au rang desquelles les problématiques de sécurité sont parmi les plus importantes. En effet, les données collectées par ces dispositifs étant des données personnelles importantes (car il s'agit de données de santé), leur protection est primordiale. Aussi, il est essentiel que les dispositifs connectés mettent en œuvre un minimum de mécanismes de sécurité afin de convaincre les utilisateurs d'adopter largement des solutions basées sur l'IoT.
Ainsi, en prenant en compte les contraintes d'énergie, de mémoire et de calcul des objets intelligents qui rendent l'utilisation des protocoles de sécurité traditionnels inadéquats, le défi majeur de l'IoT, en particulier dans le domaine médical, est de disposer de solutions de sécurité adaptées pour sécuriser les communications. Si la sécurisation des objets eux-mêmes est aussi à prendre en compte, et nous en discuterons plus loin, c'est bien la sécurité des communication qui est le

principal frein à son large déploiement.

L'objectif de ce chapitre est de présenter un état de l'art sur les problématiques de sécurité des communications de l'Internet des Objets en général et dans le domaine médical en particulier, mais également de présenter des outils que nous utiliserons dans les prochains chapitres. Ainsi, nous commencerons par introduire les menaces et attaques contre la sécurité des communications dans l'IoT. Puis, nous rappellerons brièvement les définitions de quelques propriétés de sécurité souhaitées pour se protéger contre les menaces. Nous présenterons ensuite des mécanismes cryptographiques que nous utiliserons dans les chapitres suivants pour mettre en œuvre des solutions adaptées pour la sécurité des communications. Pour finir, nous intéresserons particulièrement aux protocoles d'authentification existants dans la littérature pour les modes de communication M2M et M2C.

2.2 Les menaces et attaques

La sécurité représentant un élément essentiel pour une large adoption des solutions utilisant l'IoT, il est important d'identifier les menaces et attaques pouvant cibler les périphériques utilisés. Les menaces sont de deux ordres : celles contre les périphériques IoT eux-mêmes et celles contre les communications. Dans cette thèse, nous ne considérerons que celles contre les communications. Ce n'est pas que nous jugeons celles contre les périphériques anecdotiques. Nous sommes convaincus qu'il est important de les adresser puisque c'est eux qui collectent les données initiales et/ou qui possèdent des secrets pour la sécurisation des communications. D'ailleurs de nombreux chercheurs travaillent pour proposer des mécanismes utilisant des composants matériels sécurisés [109] ou encore des enclaves matérielles et/ou logicielles afin de protéger données et secrets cryptographiques. D'autres proposent des mécanismes d'attestations [110, 111, 112] qui permettent de vérifier qu'un dispositif IoT n'a pas été compromis. Toutefois, comme dans une thèse il est impossible d'adresser tous les aspects, nous avons choisi de nous concentrer sur la sécurité des communications. En effet, la compromission physique d'un objet a souvent moins d'impact sur un système IoT que la compromission des communications entre les différents entités. En somme, dans notre modèle d'attaquant, nous faisons l'hypothèse que l'objet est soit physiquement sécurisé, soit qu'il est dans un environnement contrôlé, ce qui est raisonnable dans le contexte médical (c'est-à-dire qu'il n'est pas *adverse* au contraire de capteurs environnementaux déployés dans une forêt par exemple). Dans le cadre de la thèse, les objets intelligents utilisés sont quasi-systématiquement dans un domicile ou dans une structure médicale et y accéder physiquement est assez difficile. En revanche, accéder à ces objets à distance au travers des communications depuis l'extérieur est tout à fait possible.

Maintenant que nous avons motivé pourquoi nous nous intéresserons essentiellement à la sécurité des communications, nous allons présenter les grandes attaques. Elles peuvent être classées en deux types : attaques passives et attaques actives.

2.2.1 Les attaques passives

Une attaque passive consiste à surveiller, capturer ou à utiliser les informations d'un système sans que cela n'ait de répercussion ou modification sur son bon fonctionnement ou sur les données échangées. En effet, ce type d'attaque n'a aucun impact sur les communications entre les entités du système. Ces attaques sont souvent des étapes préliminaires à une attaque active. Conformément à notre modèle de l'attaquant énoncé plus haut, nous allons présenter deux attaques : l'écoute passive et l'attaque relais. En effet, les attaques par canaux auxilaires (« side channel ») [113] qui sont des attaques passives qui s'intéressent à récupérer les signatures électromagnétiques, la consommation électrique, etc. d'un objet en fonctionnement pour extraire la plupart du temps des secrets cryptographiques, sont hors de notre modèle.

2.2.1.1 L'écoute passive (« Eavesdropping »)

L'écoute passive consiste à surveiller et à écouter sans modifier les données ou le fonctionnement du système. Lors d'une attaque de type « eavesdropping », l'attaquant écoute passivement les communications pour accéder à des informations sensibles envoyées ou reçues sur le réseau. Les informations peuvent être en claires ou chiffrées, si des mécanismes cryptographiques ont été utilisés. Les informations intéressantes peuvent être les identifiants des dispositifs connectés ou encore ceux des services ou des utilisateurs, voire des données (par exemple une mesure d'une constante de santé). Même chiffrés, ces identifiants ou données peuvent être intéressants pour un attaquant car si aucun mécanisme permettant de les faire évoluer au cours du temps n'a pas été prévu alors l'écoute passive peut permettre d'établir des corrélations et potentiellement d'impacter le respect de la vie privée. Comme on le constate sur la Figure 2.1 qui illustre un intrus écoutant secrètement l'échange entre un utilisateur légitime (la clé de voiture) et sa voiture, les attaques par écoute passive sont difficiles à détecter (et même parfois impossible à empêcher) en particulier dans le contexte de communications sans fil omnidirectionnelles. En effet, l'attaquant peut être placé n'importe où à partir du moment où il peut recevoir le signal.

2.2.1.2 Attaque relai

Dans une attaque relai, l'attaquant intercepte un message émis par une des deux entités communicantes et le relaie à la seconde alors que cette dernière est hors de la portée de communi-

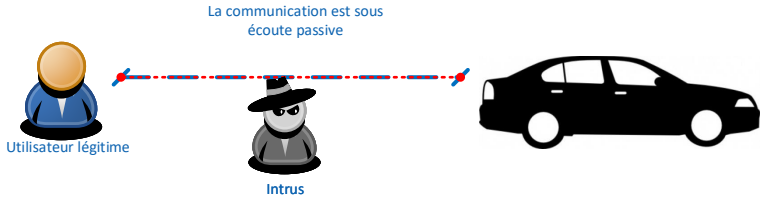


FIGURE 2.1 – Écoute passive

cation de l'émetteur original. L'attaquant va ainsi relayer au niveau physique tous les messages émis par chacune des deux entités, leur faisant croire qu'elles sont à portée de communication. Pour cela, l'attaquant peut être dans la portée de communication des deux parties sans que pour autant celles-ci ne soient capables de communiquer directement comme dans le scénario de la Figure 2.2 ; il peut aussi utiliser des dispositifs placés à proximité des deux parties pour relayer leurs messages via un second canal de communication (un canal longue distance permettant d'interconnecter ses deux dispositifs) comme dans le scénario de la Figure 2.3.

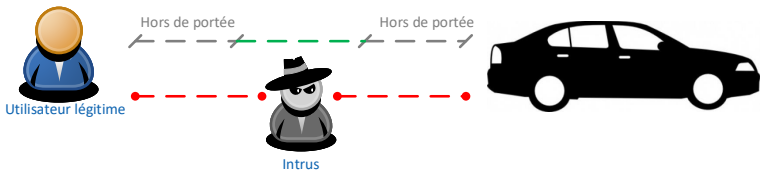


FIGURE 2.2 – Attaque relai - scénario 1

Sur le scénario de la Figure 2.2, il est possible d'imaginer une voiture dont les portières se déverrouillent lorsque l'utilisateur (plus exactement sa clé communicante) est à une distance assez proche (mais qui est ici trop grande pour que cela soit le cas). Pourtant la présence d'un attaquant qui se situe dans la porté de communications des deux entités (voiture et clé) peut en relayant tous les messages leur faire croire qu'elles sont à proximité, ce qui déclenche l'ouverture des portes.

Sur le scénario de la Figure 2.3, l'attaquant va recréer le sentiment de proximité entre les entités pourtant distantes (par exemple utilisateur ayant sa clé dans sa poche à son bureau, et voiture sur un parking) en utilisant deux dispositifs qui vont communiquer entre eux via un second canal longue distance (par exemple GSM) afin de relayer la totalité des messages émis par les entités légitimes ; ce qui déclenche là aussi l'ouverture des portes.

Cette attaque est très puissante car comme elle travaille au niveau de la couche physique, elle passe outre les mécanismes cryptographiques utilisés. Elle a longtemps été utilisée dans les systèmes RFID passifs et actifs et aujourd'hui sur les objets de l'IoT. À titre d'information, puisque nous ne les détaillerons pas par la suite, seuls les protocoles limiteurs de distance (« distance-

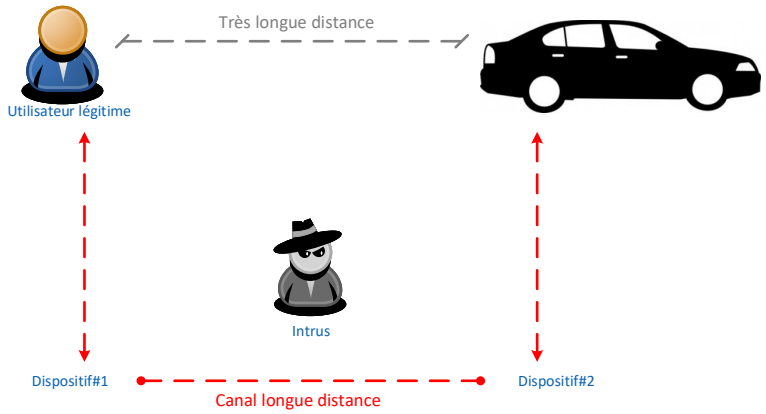


FIGURE 2.3 – Attaque relai - scénario 2

bounding ») [114] sont capables de détecter ce type d'attaque.

2.2.2 Les attaques actives

Une attaque active consiste à capturer et à modifier les informations d'un système, voire même à en injecter. Aussi dans le contexte de la sécurité des communications, il faut considérer que l'attaquant à un accès total au canal de communication. Il peut donc ajouter, modifier ou supprimer les messages. C'est typiquement le modèle de Dolev-Yao [115]. Ainsi, il peut tenter d'envoyer à une entité du système un message en essayant d'usurper l'identité d'une autre entité afin d'avoir, par exemple, accès direct à des données personnelles. Parmi les attaques actives, nous présenterons l'attaque de l'homme du milieu MITM (« Man in the Middle ») et la désynchronisation car elles sont utilisées ultérieurement dans la thèse.

2.2.2.1 L'attaque de l'homme du milieu MITM (« Man in the Middle »)

Dans une attaque du type homme du milieu, l'attaquant intercepte les messages sur un support de communication entre deux parties légitimes. Ce type d'attaque est souvent utilisé lors de phases d'authentification afin que l'attaquant puisse ultérieurement analyser le contenu des futurs échanges. Les attaques MITM consiste alors à faire croire aux entités communicantes qu'elles échangent directement alors qu'en réalité chaque partie communique avec un intermédiaire (l'attaquant) qui pourra ainsi contrôler les conversations entre deux parties. Contrairement à l'attaque relai dans laquelle l'attaquant ne modifiait aucune information, dans l'attaque MITM, il injecte et/ou modifie des messages. La Figure 2.4 illustre le positionnement de l'attaquant (sur le canal de communication) dans une attaque MITM où la voiture pense parler à l'utilisateur (la clé) et l'utilisateur pense parler à la voiture alors qu'en fait les deux dialoguent avec l'attaquant.

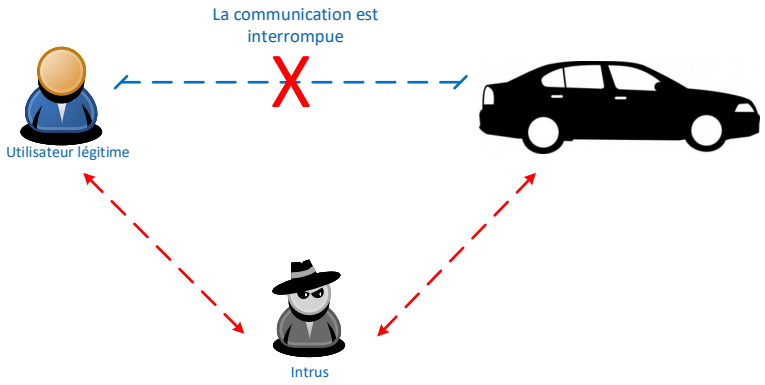


FIGURE 2.4 – Homme du milieu

2.2.2.2 La désynchronisation

L'objectif de l'attaque de désynchronisation est de faire en sorte que deux entités communicantes ne puissent plus s'authentifier. L'attaquant va fabriquer ou rejouer des messages de façon à ceque les valeurs secrètes partagées par les entités se retrouvent désynchronisées; s'il n'existe pas de mécanismes de resynchronisation alors toute communication future (sécurisée) sera impossible. C'est une forme de déni de service DoS (« Deny of Service »).

2.3 Les propriétés de sécurité

Face aux menaces, il est fréquent de définir les propriétés de sécurité à assurer dans un système. Lors de conception d'une solution de sécurité visant à pallier les menaces identifiées, une fois les propriétés de sécurité souhaitées exprimées, celles-ci se nomment sous le terme d'exigences de sécurité. Ces propriétés seront utilisées dans les solutions de sécurité que nous proposerons dans la suite.

2.3.1 L'authentification

L'authentification représente souvent la première barrière de sécurité pour empêcher une personne tierce et non autorisée d'accéder aux données des dispositifs intelligents d'un système IoT. S'il y a plusieurs façons de réaliser l'authentification, dans un système de communications elle utilise souvent des mécanismes cryptographiques comme la signature numérique qui sera présentée plus loin. Dans un système, souvent, il est nécessaire que toutes les entités d'un système prouvent qu'elles sont bien qui elles prétendent être. Ce processus d'authentification peut être à sens unique et dans ce cas l'entité cherchant à s'authentifier est appelée *prouveur* (car elle cherche

à prouver son identité) alors que celle auprès de qui elle initie le processus d'authentification (c'est-à-dire de vérification de la preuve) s'appelle *vérifieur*. Mais ce processus peut également être à double-sens, c'est-à-dire que chaque entité vient prouver à l'autre qu'elle est bien qui elle prétend et dans ce cas l'authentification est dite *mutuelle*.

2.3.2 La confidentialité

La confidentialité des données est la garantie que seules les entités autorisées peuvent lire les données pour lesquelles elles ont les droits. Dans un système IoT, et en particulier dans le contexte des communications, la confidentialité revient à assurer la protection des données échangées contre l'interception par des personnes non autorisées en utilisant des mécanismes cryptographiques.

2.3.3 L'anonymat

L'anonymat est une forme particulière de confidentialité, puisqu'elle consiste généralement à protéger une donnée particulière : l'identité. Dans un système IoT, l'anonymat consiste à garder secrète ou indiscernable (c'est-à-dire en pratique semblable à un aléa) toute information permettant d'identifier une entité communicante. En effet, une entité est anonyme si elle n'est pas identifiable. Lors d'une communication entre un émetteur et un récepteur, l'anonymat est garanti si l'analyse d'un message ne permet pas d'identifier son émetteur et/ou son récepteur.

2.3.4 L'intégrité

L'intégrité est le pendant de la confidentialité. L'intégrité des données est la garantie que seules les entités autorisées peuvent écrire les données pour lesquelles elles ont les droits. Comme un attaquant peut modifier les données lors de la transmission, dans un système IoT, assurer l'intégrité des données consiste à ajouter aux données des informations supplémentaires qui permettront au destinataire de vérifier si elles n'ont subi aucune modification (altération ou destruction volontaire ou accidentelle) lors de la transmission entre la source et la destination. Là encore, cette propriété est souvent assurée en utilisant des mécanismes cryptographiques (même si pour des simples erreurs de transmission des codes correcteurs ou détecteurs d'erreurs peuvent suffire).

2.3.5 La disponibilité

C'est la garantie de l'accès à un service ou à des ressources afin de maintenir le bon fonctionnement du système IoT. Assurer la disponibilité, c'est donc construire le système afin de minimiser l'impact d'un déni de service DoS qu'essayerait de provoquer un attaquant pour empêcher les utilisateurs légitimes d'y accéder.

2.3.6 La non-répudiation

La non-répudiation est la garantie qu'une action a été effectuée par une partie. En particulier dans le contexte des communications, la non-répudiation est l'assurance (c'est-à-dire la preuve) qu'un émetteur ne peut nier avoir transmis des messages; on parle de preuve de l'origine. Là, aussi, c'est souvent le mécanisme cryptographique de signature numérique qui est utilisé en pratique pour assurer cette propriété.

2.4 Le respect de la vie privée

La vie privée (intimité ou privacy en anglais) constitue une exigence fondamentale dans l'IoT, du fait des informations personnelles sensibles manipulées. Les données des utilisateurs de cette technologie peuvent être liées à leurs mouvements, habitudes et leurs interactions avec d'autres personnes/objets. De ce fait, la protection des informations sensibles doit être mise en œuvre au niveau du dispositif IoT ainsi que pendant leur stockage, leur communication et leur traitement.

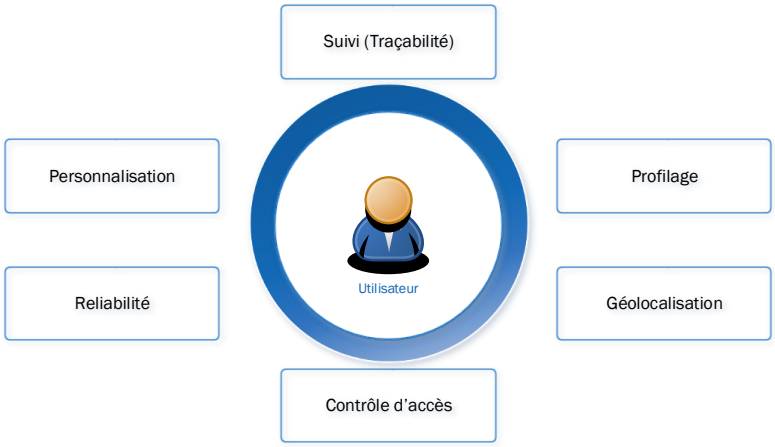


FIGURE 2.5 – Respect de la vie privée de l'utilisateur de l'IoT

L'environnement de l'IoT se caractérise par son omniprésence parfois même transparente grâce à la miniaturisation des dispositifs (l'utilisateur oublie qu'il est entouré d'objets connectés). De plus, comme ces objets intelligents sont connectés à internet, ils peuvent communiquer et échanger les données récoltées en toute transparence, parfois même sans que leurs propriétaires en aient connaissance, comme le montre la Figure 2.5, ce qui rend le respect de la vie privée des utilisateurs de l'IoT un sujet sensible auquel de nombreux travaux de recherche se sont intéressées [12, 116, 117].

Face au grand nombre d'objets connectés ambiants, portés ou incorporés, l'identification de ces objets d'une manière anonyme (seules les entités légitimes peuvent identifier les objets connectés)

devient primordiale dans le cadre du respect de la vie privée des utilisateurs de cette technologie. L'identification consiste à associer une identité pertinente à une personne et à ses données. Cet identifiant peut être par exemple un nom, une adresse, une empreinte digitale, etc. La technologie RFID a indéniablement participé à la réalisation de la vision de l'IoT, en permettant l'identification passive automatique à bas coût via les tags passifs attachés à des objets/personnes. Néanmoins, la propagation de l'utilisation de cette technologie n'est pas sans risque, car elle engendre d'autres des menaces de sécurité, comme l'identification automatique et le suivi des personnes à l'aide d'étiquettes cachées (par exemple dans des vêtements [118]). La menace contre l'identification est le fait d'associer une identité à un contexte spécifique violant ainsi la vie privée de l'utilisateur à son insu comme cela est possible avec l'utilisation des caméras de surveillances connectées (l'identification automatique des individus à partir des images des caméras est déjà une réalité) [119].

Grâce à l'utilisation des réseaux de capteurs, les objets du quotidien sont complétés par des capacités de détection, de traitement et de communication permettant de construire des réseaux d'objets interconnectés. En dépit des avantages qu'ils peuvent offrir à leurs propriétaires, le fait que ces objets soient augmentés d'une connexion à internet augmente les risques de piratage [120]. De plus, en cas de perte, de vol ou même de piratage, des données importantes et personnelles peuvent être touchées. En effet, la plupart des capteurs utilisés ne présentent aucun mécanisme de sécurité, il est tout à fait possible pour un attaquant de les interroger et d'obtenir des informations sur le propriétaire de l'objet. Par exemple, l'utilisation d'une alarme domestique connectée ou bien d'un accès à une maison intelligente par une serrure connectée accentue les menaces; selon des mécanismes de sécurité implémentés, les vols des biens peuvent être plus faciles à réaliser.

D'autres formes d'atteintes à la vie privée sont la localisation et le suivi anonyme des personnes. Cette menace consiqte à déterminer et d'enregistrer les différentes localisations physiques d'une personne dans l'espace et à suivre ses déplacement dans le temps à son insu. Ce suivi nécessite l'utilisation d'un identifiant unique afin de lier des localisations consécutives à un seul individu. Un tel suivi est possible par différents moyens, tels que l'utilisation d'un système de positionnement global GPS (« Global Positioning System ») [121] ou bien l'utilisation de la localisation via un téléphone mobile qui permettent non seulement de savoir où vit la personne surveillée mais également de connaître ses habitudes.

Un autre aspect d'atteinte à la vie privée est le profilage, résultant de la capacité d'identifier un objet/individu. Autrement dit, le profilage désigne la menace d'amasser ou de collecter des informations sur des personnes dans le but d'établir leurs profils, pour en déduire des intérêts par corrélation avec d'autres profils et bases de données. L'intégration des données d'une variété de bases

de données permet de construire des groupes de profils raffinés selon un ensemble d'attributs représentant d'une manière adéquate et détaillée les catégories (ou communautés) des personnes. De ce fait, le groupe peut être composé de personnes qui appartiennent à une même classe d'étudiants, ou bien des adeptes d'une religion ou des membres d'une association. Le groupe peut également être constitué d'une catégorie de personnes qui n'ont aucun lien entre elles, outre le fait que le profilage les a définies comme une catégorie [122]. Par conséquent, dans le but de garantir le respect des données privées d'un individu, il est nécessaire de traiter les données à caractère personnel sans possibilité de déduction à l'égard du propriétaire de l'objet intelligent.
La vie privée des utilisateurs et la protection de leurs données sont identifiées parmi les défis les plus importants de l'IoT. La protection de l'information sensible doit être mise en œuvre au niveau du dispositif ainsi que pendant la récolte, le stockage, la communication et le traitement des données. Dans tous les cas, l'utilisateur doit être notifié du processus qui accède à ses données personnelles et il doit être en mesure de se retirer à tout moment. L'individu devrait également avoir la possibilité de contrôler quelle information à son sujet et jusqu'à quel niveau de détails il est prêt à divulguer et à partager [123, 10].

2.4.1 Le respect de la vie privée et la santé

Le respect de la vie privée est considéré comme une préoccupation majeure pour l'adoption des technologies de l'IoT dans le domaine médical. Lors de l'utilisation de capteurs sur des patients à des fins de surveillance médicale, il est essentiel d'établir un contrôle sur l'accès aux données collectées : le patient et son médecin traitant doivent y avoir accès, et il serait souhaitable que d'autres personnes tels que les services d'urgence ou les personnes autorisées puissent y accéder au besoin sachant que tout autre accès doit impérativement être refusé. Dans les applications de santé basées sur l'IoT, les dispositifs intelligents collectent et transmettent les données sensibles du patient. Un adversaire peut écouter (eavesdropping) la communication et déduire des informations critiques (par exemple qui communique avec qui, ou quel est le volume des données échangées) [124, 125, 87], ce qui peut causer de graves dommages au patient car l'adversaire peut utiliser les données acquises à de nombreuses fins illégales pouvant même avoir un impact direct sur la vie des personnes. Par exemple, le piratage d'une pompe à insuline ou d'un pacemaker peut créer une nuisance directe à son propriétaire, voire le tuer (injection d'une dose mortelle d'insuline ou arrêt du pacemaker). Un intrus peut aussi intercepter et modifier les données d'un patient ce qui peut mener à un mauvais diagnostic (basé sur les données erronées) et donc d'entraîner des conséquences graves.
Les mauvaises pratiques et le manque de moyens des hôpitaux font que dans certains cas ces derniers ne disposent pas des budgets nécessaires pour s'équiper de matériel fiable (l'absence de

contrôle de sécurité des objets connectés pendant leur fabrication) ou encore pour former leur personnel à la cyber sécurité [126] en santé. Par conséquent, toute la panoplie de la cyber sécurité qui correspond aux vulnérabilités, menaces, intrusions, expositions et attaques se déclinent dans le domaine médical tant sur les objets eux-mêmes que sur leurs services. Ainsi par exemple dans le contexte de l'utilisation d'une montre connectée pour la surveillance continue du rythme cardiaque, la détection de chute, et en cas de nécessité, pour déclencher un appel d'urgence, le piratage de l'objet conduira à diffuser des informations erronées. Il est en effet possible de déclencher des signaux intempestifs en donnant l'illusion d'une chute en changeant les les données issues de l'accéléromètre de la montre. Ce type d'attaque pourrait perturber les services d'assistance et d'intervention d'urgence. D'une façon plus générale, le déclenchement systématique d'appels d'urgence pourrait conduire à des attaques DoS sur les services d'intervention d'urgence.
Au cours des dernières années, le nombre d'attaques de sécurité qu'ont subi principalement les hôpitaux n'a cessé de croître. Parmis elles, plusieurs visaient à compromettre des dispositifs connectés individuels, notamment les systèmes d'administration de médicaments, des implants électroniques pour la santé, des pompes à insuline et des stimulateurs cardiaques. En particulier, l'attaque sur des dispositifs médicaux Hijack (MEDJACK : Medical Device Hijack) [127], découverte pour la première fois par Trend Micro, a eu des répercussions sur les analyseurs de gaz sanguins, les appareils de tomographie par ordinateur, les systèmes d'imagerie par résonance magnétique et les appareils à rayons X. Dans [128], les auteurs ont démontré que les implants cardiaques ICDs (« Implantable Cardiac Defibrillators ») contiennent des défauts de sécurité potentiellement mortels. Ils ont mis en évidence de graves vulnérabilités dans les protocoles sans fil d'un neurostimulateur implantable, leur permettant de réaliser une gamme d'attaques de sécurité sans nécessiter d'accès physique aux dispositifs des patients.
Compte tenu des risques sur la vie privée, les applications IoT de santé doivent être ouvertes et transparentes pour les utilisateurs (les patients et les professionnels de santé), de plus elles doivent indiquer les raisons et les besoins pour lesquels les informations de santé sont collectées. La conservation des informations nécessaires et leur mise à jour doit être assurée et protégée. L'utilisation des mécanismes de sécurité adaptés aux besoin des applications médicales devient primordiale afin de permettre aux adeptes de l'IoT de gérer leurs propres données de santé tout en bénéficiant d'une vie autonome.

2.5 Les mécanismes de sécurité

Dans cette section, nous allons présenter les grands types de mécanismes de sécurité que nous utiliserons dans les chapitres suivants. De manière générale, les mécanismes de sécurité visent à

protéger l'accès aux biens (c'est-à-dire aux données et ressources) d'un système contre les menaces de sécurité. Ainsi, l'utilisation des mécanismes de sécurité permet de mettre en œuvre les services de sécurité afin d'empêcher la divulgation et/ou la modification non-autorisée des données et/ou l'accès non-autorisé aux ressources. Par exemple, pour assurer l'accès aux ressources aux seuls utilisateurs autorisés, un service d'authentification peut être mis en place. Celui-ci peut se réaliser via l'utilisation de différents mécanismes de sécurité : présentation d'un couple identifiant et mot de passe, technique challenge-réponse utilisant soit du chiffrement symétrique, soit de signature numérique, etc. Parmi ces quelques techniques toutes ont des avantages mais aussi des inconvénients. Si la comparaison d'un simple couple identifiant/mot de passe est très simple à implémenter, son niveau de sécurité est plutôt faible. À l'inverse, la signature numérique est plus complexe à implémenter mais néanmoins elle fournit des propriétés supplémentaires comme la garantie d'intégrité et la non répudiation.

Clairement dans la suite, nous ne souhaitons pas être exhaustifs mais seulement donner les principaux mécanismes cryptographiques que nous allons utiliser : les différents types de fonctions de hachage et les algorithmes de chiffrement (symétriques ou non) et de signature. Il est à noter qu'il s'agit bien des mécanismes et non de leur mise en œuvre effective dans le monde réel, car celle-ci nécessite souvent des raffinements supplémentaires (choix des algorithmes, des tailles des clés, etc.).

2.5.1 Les fonctions de hachage

Les fonctions de hachage sont des fonctions qui prennent une entrée de longueur arbitraire et la compressent en un résultat d'une longueur fixe, que l'on nomme « haché » ou « condensé ». Suivant la fonction de hachage utilisée, la taille de la sortie produite diffère. Toutefois cette taille est fixe pour une fonction de hachage donnée (quelque soit l'entrée donnée). De la même façon, selon la fonction de hachage utilisée, pour une valeur d'entrée donnée, la valeur de sortie produite changera, mais pour une fonction de hachage donnée et pour une valeur d'entrée donnée, c'est toujours le même haché qui sera produit.

Les fonctions de hachage sont largement utilisées dans la conception des protocoles cryptographiques, aussi bien que ceux utilisant de la cryptographie symétrique qu'asymétrique. Ainsi, en cryptographie, les fonctions de hachage sont souvent utilisées pour l'authentification, les signatures numériques et les codes d'authentification de messages.

Les principales caractéristiques que doit posséder une fonction de hachage cryptographique sont :

1. D'être rapide à calculer. Elle doit retourner la valeur du haché rapidement car elle est fréquemment sollicitée dans la mise en œuvre de protocoles.

2. D'être non réversible (on dit aussi « unidirectionnelle »). Par définition, il ne doit pas être possible de reconstruire le message initial à partir d'un haché donné.
3. De résister à la falsification. Toute modification d'un message initial entraîne la modification du résultat de hachage (c'est-à-dire, les hachés sont différents).
4. De résister aux collisions. Par définition, il ne doit pas être possible de trouver deux messages différents qui produisent le même haché.

Introduisons quelques notations afin de formaliser les fonctions de hachage que nous utiliserons dans la suite :

- h, une fonction de hachage ;
- X, l'entrée de la fonction (l'information à hacher) ;
- Y, le haché, c'est-à-dire, l'image de X par la fonction h.

Essentiellement, les fonctions de hachage cryptographiques se divisent en deux catégories :

- Les fonctions de hachage permettant de réaliser des Codes de Détection de Manipulation, MDC (« Manipulation Detection Code ») qui s'expriment ainsi $Y = h(X)$. Parmi ces fonctions, ils existent :
  - des fonctions de hachage à sens unique, OWHF (« One-Way Hash Function »), pour lesquelles il est difficile de retrouver le message initial X à partir du haché Y et de h et également de trouver X' tel que $h(X) = h(X')$ avec $X' \neq X$ à partir de X et $Y = h(X)$
  - des fonctions de hachage résistantes aux collisions, CRHF (« Collision Resistant Hash Function »), pour lesquelles il est difficile de trouver deux messages distincts X et X' (c'est-à-dire $X' \neq X$) tels que $h(X) = h(X')$ [129, 130, 131, 132].
- Les fonctions de hachage permettant de réaliser des Codes d'Authentification de Message, MAC (« Message Authentication Code »), qui requièrent une clé secrète k pour le calcul du haché. Ces fonctions qui peuvent être implémentées soit en utilisant un algorithme de chiffrement symétrique de bloc en mode CBC (« Cipher Block Chaining ») comme AES, soit en utilisant des fonctions MDC. Ainsi, si $Y = h_k(X)$, connaissant, Y, X et h, k doit être difficile à calculer et de la même manière, connaissant Y, k et h, X doit être difficile à calculer.

L'utilisation des fonctions de hachage en sécurité permet de représenter les données au travers d'une image authentique plus petite et de longueur fixe que l'on nomme aussi *empreinte*. D'une certaine façon, l'utilisation d'une fonction de hachage permet d'assurer l'intégrité des données et indirectement de les authentifier. Souvent, les schémas de signatures numériques utilisent des fonctions de hachage de la catégorie MDC pour compresser les informations à signer en un haché de longueur fixe. Ainsi, l'empreinte sera signée, et non l'information originale (qui peut être

très grosse) afin de minimiser le temps de calcul. Afin de vérifier la signature lors de la transmission d'un message, le résultat du processus de vérification de la signature sera comparé au haché calculé (avec la même fonction de hachage) sur les données reçues. Si ces résultats concordent le message reçu est bien celui qui a été signé et transmis.

Nous utilisons les fonctions de hachage dans les solutions proposées dans les chapitres 3, 4 et 5.

2.5.2 Les mécanismes de chiffrement

Pour réaliser le chiffrement, processus de transformation d'un texte clair en message chiffré, il est possible d'utiliser la cryptographie symétrique ou la cryptographie asymétrique. L'objectif du chiffrement est soit de protéger la confidentialité d'une information échangée entre des entités communicantes lors de son transfert, soit de protéger la confidentialité d'une information lors de son stockage (plutôt qu'un échange dans l'espace, le stockage peut être vu comme un échange dans le temps entre un utilisateur – il écrit son information – et lui-même – il lit son information). L'opération inverse du chiffrement est le déchiffrement. Selon le type de cryptographie utilisée, la façon d'utiliser les clés diffère.

2.5.2.1 Le chiffrement symétrique

Afin d'échanger des données de façon sécurisée, avec un algorithme de chiffrement symétrique, l'émetteur et le récepteur (ou les récepteurs) doivent utiliser un secret partagé (c'est-à-dire une clé). La notion de symétrie provient du fait que c'est la même clé, qualifiée de secrète, que les entités communicantes utilisent à la fois pour chiffrer (opération de transformation du message clair en message chiffré) et pour déchiffrer (opération de transformation du message chiffré en message clair). La clé doit rester secrète pour les entités du système n'ayant pas les droits (par contre, elle peut être partagée au sein d'un groupe si cela est nécessaire et elle devient une clé de groupe – toutefois, une fois la clé de groupe partagée, il est difficile d'exclure un membre du groupe car il faut repartager la clé) [133, 134]. La problématique principale du chiffrement symétrique est qu'il nécessite un partage préalablement à l'échange des données chiffrées du secret partagé (c'est-à-dire de la clé secrète) via un canal sûr (c'est-à-dire un canal sur lequel un attaquant n'a pas le contrôle).

On pourra aussi noter qu'il est souvent pertinent dans les protocoles utilisant les algorithmes de chiffrement symétrique de faire évoluer la clé secrète utilisée lors des échanges à différents moments dans le temps. L'évolution de la clé doit, autant que possible, permettre d'assurer les propriétés de « Perfect Forward Secrecy » et « Perfect Backward Secrecy » qui garantissent que la découverte par un adversaire de la clé secrète ne compromet pas la confidentialité des communi-

cations respectivement, passées et futures.

Parmi les algorithmes de chiffrement symétrique que nous pourrions utiliser, il y a le standard AES [135] ou encore le masque jetable (one time pad) qui utilise la propriété du `xor`.

Nous utilisons le chiffrement symétrique dans des solutions proposées aux chapitres 4 et 5.

2.5.2.2 Le chiffrement asymétrique

Pour réaliser le chiffrement asymétrique, chaque entité communicante doit posséder deux clés cryptographiques : une clé privée (connue seulement par l'entité qui la possède) et une clé publique (accessible par tous).

Dans un tel schéma, une tierce partie de confiance peut agir comme une autorité de certification CA (« Certificate Authority ») soit pour signer les clés publiques afin de prouver leur authenticité, soit pour générer les paires de clés (privée/publique) pour chaque entité et pour fournir les clés publiques à tous. Si l'on se place dans ce dernier cas, lorsque une entité désire envoyer un message sécurisé à une autre, elle chiffre son message avec la clé publique du destinataire, clé récupérée auprès de la CA. L'entité qui reçoit le message le déchiffre en utilisant sa propre clé privée [136, 137]. Tout le monde peut chiffrer pour une entité avec la clé publique mais seule le possesseur de la clé secrète peut déchiffrer le message.

Parmi les algorithmes de chiffrements asymétriques que nous pourrions utiliser, il y a le standard RSA [138] ou encore un cryptosystème reposant sur la cryptographie sur les courbes elliptiques, ECC.

Nous utilisons le chiffrement asymétrique dans une solution proposée au chapitre 3.

2.5.3 Les cryptosystèmes à clé publique

Les cryptosystèmes à clé publique sont complémentaires de ceux à clés symétriques qui permettent essentiellement d'assurer le chiffrement efficace. Par exemple, ils permettent d'assurer le processus d'authentification afin d'initier la mise en place de clés de session (c'est-à-dire des clés symétriques) pour protéger le canal de communication. Ainsi, nous allons présenter brièvement le fonctionnement de RSA, premier cryptosystème de ce type, pour ensuite nous attarder sur la cryptographie basée sur les courbes elliptiques que nous avons largement utilisée car elle est particulièrement adaptée aux environnements contraints comme ceux qui sont étudiés dans cette thèse.

2.5.3.1 RSA (« Rivest-Shamir-Adleman »)

L'algorithme RSA a été conçu par Ron Rivest, Adi Shamir et Leonard Adleman en 1978. C'est un cryptosystème asymétrique à clé publique permettant l'échange de clés, le chiffrement de blocs

de données et la signature numérique [139, 140].

Les clés publique et privée sont générées à partir de deux nombres premiers distincts. Ces deux clés différentes sont utilisées pour le chiffrement et le déchiffrement et pour la signature et le processus de vérification de la signature.

La sécurité du cryptosystème RSA repose sur la difficulté à factoriser des grands nombres (c'est-à-dire qu'il n'existe pas d'algorithme rapide de complexité polynomiale sur des ordinateurs classiques).

La procédure de génération de clés est simple :

- Choisir deux grands nombres premiers aléatoires p et q tels que $p \neq q$
- Calculer $n = p \cdot q$
- Calculer $\phi(n) = (p-1) \cdot (q-1)$
- Choisir un entier naturel e, l'exposant public, tel que $1 < e < \phi(n)$ et $pgcd(e, \phi(n)) = 1$
- Calculer d, l'exposant privé, pour satisfaire la relation $d \cdot e \equiv 1 \mod \phi(n)$
- La clé publique est (n, e) et la clé privée est (n, d). Il faut conserver toutes les valeurs d, p et q secrètes.

Afin d'échanger des données de façon confidentielle, l'expéditeur chiffre le message à l'aide de la clé publique du destinataire. Lorsque le message est reçu, le destinataire peut le déchiffrer à l'aide de sa propre clé privée. Le processus de chiffrement et de déchiffrement est le suivant :

- Chiffrement d'un message
  - M est un entier naturel représentant un message clair, et M est choisi tel que $M < n$.
  - Le message chiffré C est alors calculé ainsi : $C \equiv M^e \mod n$.
- Déchiffrement d'un message
  - Déchiffrer le cryptogramme C pour obtenir M consiste à calculer $M \equiv C^d \mod n$.

Pour signer un message, c'est-à-dire prouver que le message provient bien de l'expéditeur, ce dernier va hacher le message à authentifier à l'aide d'une fonction de hachage pour obtenir une empreinte qu'il va chiffrer (on parle ici de *signer*) avec sa propre clé privée afin d'obtenir la signature. À la réception du message et de la signature, le destinataire va utiliser la clé publique de l'émetteur pour déchiffrer l'empreinte (contenue dans la signature) et la comparer à celle qu'il obtient en appliquant la même fonction de hachage sur le message reçu (c'est le processus de vérification de la signature). Si les empreintes concordent alors le message reçu est bien celui envoyé par l'émetteur. Le processus de signature et de vérification est le suivant :

- Signature d'un message
  - M est le message à signer et $h(x)$ est une fonction de hachage.

— La signature σ se calcule ainsi : $\sigma \equiv h(M)^d \mod n$.

- Vérification de la signature d'un message
  - — M est le message reçu et σ est la signature reçue associée au message.
  - — Si vérification de $h(M) \stackrel{?}{=} \sigma^e \mod n$ donne `vrai` alors la signature est correcte et fausse sinon.

2.5.3.2 Les courbes elliptiques

La cryptographie sur les courbes elliptiques ECC (« Elliptic Curve Crypthography ») est une approche de chiffrement à clé publique utilisant plusieurs propriétés des courbes elliptiques [141]. La sécurité de ce cryptosystème repose sur la difficulté de calculer le logarithme discret dans le groupe des points rationnels de certaines courbes elliptiques. En effet, le chiffrement ECC est basé sur l'utilisation d'une fonction unidirectionnelle qui produit pour chaque entrée particulière une sortie unique qu'il est facile à calculer mais pour laquelle il est difficile de retrouver l'entrée correspondante à partir d'une sortie particulière.

En pratique pour utiliser les ECC, toutes les entités qui participent à la communication doivent connaître les paramètres du domaine, c'est-à-dire l'ensemble (q, a, b, p, P), où p et q sont deux nombres premiers, P est un point de la courbe E appelé générateur d'ordre p.

Dans le domaine fini $\mathbb{F}_q$, la courbe elliptique E est définie par l'équation 2.1 où les constantes a et b sont des éléments de $\mathbb{F}_q$ tel que $q > 3$ et $4a^3 + 27b^2 \neq 0$.

$$y^2 = x^3 + ax + b \tag{2.1}$$

Chaque valeur de a et b donne une courbe elliptique différente. Pour un point donné $Q = (x_Q, y_Q)$, x_Q et y_Q représentent respectivement les coordonnées en x et en y de Q. Tous les points (x_Q, y_Q) qui satisferaient l'équation ci-dessus, plus un point à l'infini O_∞, se trouvent sur la courbe elliptique.

Dans les schémas de chiffrement basés sur les courbes elliptiques, la principale opération est la multiplication scalaire (« scalar multiplication » en anglais) qui correspond à l'exponentiation dans les groupes multiplicatifs. Étant donné un nombre entier k et un point $P \in \mathbb{E}$, la multiplication scalaire $k \cdot P$ est le résultat de l'addition de P à lui même k fois.

L'ordre p d'un point P sur une courbe elliptique est le plus petit entier positif tel que $p \cdot P = O_\infty$. De plus, étant donné k et l deux nombres entiers, $k \cdot P = l \cdot P$ si et seulement si $k \equiv l (mod\, p)$.

Dans ECC, une clé publique est un point S de la courbe et la clé privée est un nombre s qui sont liés par la relation : $S = s \cdot P$, c'est-à-dire que la clé publique est obtenue en multipliant la clé privée avec le point générateur P de la courbe [142]. La procédure de génération des clés est simple :

- L'entité A qui souhaite générer une paire de clés privée/publique doit utiliser les paramètres du domaine (q, a, b, p, P) associé à la courbe elliptique E et exécute les opérations suivantes. Pour générer sa paire de clés, l'entité A exécute les étapes suivantes :
  - Elle tire un nombre aléatoire k dans $\mathbb{Z}_p^*$ qui sera sa clé privée.
  - Elle calcule Q, sa clé publique telle que $Q = k \cdot P$.
  - La paire (k, Q) représente alors respectivement la clé privée et publique de l'entité.

La validation d'une clé publique consiste à vérifier si elle satisfait l'arithmétique des courbes elliptiques. Ainsi, la clé publique $Q = (x_Q, y_Q)$ peut être validée en utilisant la procédure suivante :

- Vérifier que $Q \neq O_\infty$.
- Vérifier que x_Q et y_Q sont bien des éléments de $\mathbb{F}_q$.
- Vérifier que Q se trouve bien sur la courbe elliptique définie par a et b.
- Vérifier que $p \cdot Q = O_\infty$.

Échanger une clé Diffie-Hellman (ECDH) sur les courbes elliptiques est très simple. Si A (Alice) et B (Bob) veulent sécuriser leur communication en utilisant les courbes elliptiques, après s'être mis d'accord publiquement sur la courbe elliptique utilisée $E = (q, a, b, p, P)$, ils génèrent respectivement leurs paires de clés privées/publiques (k_A, Q_A) et (k_B, Q_B).

- A envoie sa clé publique Q_A à B et, après avoir reçu celle de B, Q_B, elle calcule $R = k_A \cdot Q_B$ (c'est-à-dire $R = k_A \cdot (k_B \cdot P)$).
- B procède de même. Il envoie sa clé publique Q_B à A et, après avoir reçu celle de A, Q_A, il calcule $R = k_B \cdot Q_A$ (c'est-à-dire $R = k_B \cdot (k_A \cdot P)$).
- En effet, il s'agit bien de la même valeur R, puisque $R = k_A \cdot (k_B \cdot P) = k_B \cdot (k_A \cdot P) = (k_A \cdot k_B) \cdot P$. La clé secrète alors utilisée par A et B est par exemple la coordonnée en x du point R.

À noter qu'il est possible de faire des clés éphémères (ECDHE).

L'avantage principal du chiffrement ECC est la faible longueur des clés utilisées, sachant que les performances du chiffrement dépendent beaucoup de la taille de la clé. Ainsi, un chiffrement avec une clé de 160 bits en ECC est équivalent à un chiffrement avec une clé de 1024 bits avec RSA [143, 144] et ECC-224 bits offre une sécurité comparable à RSA-2048 bits. Ses clés plus courtes font donc d'ECC une alternative très attractive pour les dispositifs M2M dont les puissances de traitement et les ressources sont limitées.

Du point de vue de la sécurité, les algorithmes de chiffrement à base de courbes elliptiques sont aussi plus robustes que RSA car ils reposent sur la difficulté à calculer le logarithme discret sur les courbes elliptiques et non simplement sur les entiers. Ces groupes sont plus difficiles à manipuler et ils peuvent différer beaucoup les uns des autres selon les paramètres du domaine de la courbe utilisée.

Nous utilisons les ECC dans les solutions proposées aux chapitres 3 et 5.

2.5.4 La signature de Schnorr

La méthode de signature de Schnorr est une solution de signature numérique qui utilise l'algorithme d'authentification décrit par Claus-Peter Schnorr [145, 146]. La sécurité de cet algorithme repose sur la difficulté du calcul du logarithme discret.

L'algorithme d'authentification se déroule entre P (le prouveur) qui souhaite prouver sa légitimité et V (le vérifieur) qui veut la vérifier. Le fonctionnement général de l'algorithme consiste à définir les paramètres publics dont un groupe G généré par g d'ordre q (q est un nombre premier). Le prouveur P possède une clé secrète $x \in \mathbb{Z}_q^*$ et une clé publique y telle que $y = g^x$ qui est connue de V.

P prouve alors son identité à V selon les étapes suivantes.

1. P génère un entier aléatoire r tel que $r \in \mathbb{Z}_q^*$.
2. P calcule $R = g^r$ et envoie R à V.
3. V génère un entier aléatoire c tel que $c \in \mathbb{Z}_q^*$ et l'envoie à P.
4. P calcule a tel que $a = r - c \cdot x$ et l'envoie à V.
5. V vérifie si $R \stackrel{?}{=} g^a \cdot y^c$ $(= g^a \cdot (g^x)^c = g^{(r-c\cdot x)} \cdot g^{x\cdot c} = g^{r-c\cdot x+x\cdot c} = g^r = R)$. Si oui, alors P est une entité légitime qui connaît bien x.

Pour signer un message M en utilisant un schéma non-interactif, selon l'heuristique de Fiat-Shamir [147, 148] avec le protocole de Schnorr, une fonction de hachage $h : G \times \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ est introduite dans les paramètres publiques.

Le signataire doit effectuer les étapes suivantes :

1. Générer un entier aléatoire r tel que $r \in \mathbb{Z}_q^*$ et calculer $R = g^r$.
2. Calculer $c = h(R, M)$
3. Calculer $a = r - c \cdot x$.

La signature est $\sigma = (c, a)$.

Le destinataire connaissant la clé publique du signataire pourra vérifier que le message M est valide à partir de σ si $c \equiv h(R, M)$ est valide sachant que R se calcule avec $R = g^a \cdot y^c$ $(= g^{r-c\cdot x} \cdot (g^x)^c = g^{r-c\cdot x} \cdot g^{x\cdot c} = g^{r-c\cdot x+x\cdot c} = g^r)$.

Les signatures de Schnorr sont connues pour leur simplicité, leur rapidité et leur efficacité. Si elles étaient peu utilisées car protégées par des brevets, ceux-ci sont tombés et elles gagnent en popularité face à d'autres méthodes comme les signatures ECDSA (« Elliptic Curve Digital Signature Algorithm ») [149].

Nous utilisons les signatures de Schnorr dans les solutions proposées aux chapitres 3 et 5.

2.6 L'état de l'art des protocoles d'authentification existants

Comme nous l'avons présenté dans la section 1.8, il y a essentiellement deux modes de communication dans les scénarios IoT pour les applications de santé et d'AAD qui nous intéresse : M2M et M2C. Pour rappel, sur la Figure 2.6 un exemple de communication en mode M2M peut avoir lieu entre un dispositif capteur corporel du BAN échangeant avec le BCU pour lui fournir des informations de santé. Sur cette même figure, un exemple de communication en mode M2C peut avoir lieu entre le BCU et un service du cloud permettant aux médecins agréés de surveiller l'état de santé du patient.

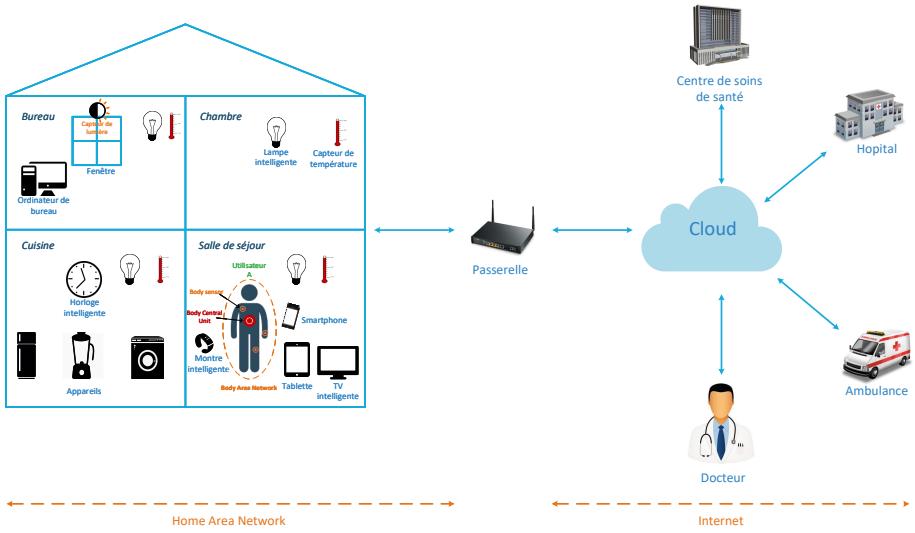


FIGURE 2.6 – Applications de santé et d'AAD dans un scénario d'une maison intelligente

Compte tenu de l'importance des données manipulées (données de santé) aux travers de ces deux exemples simples, il est évidemment essentiel de disposer de solutions de sécurité pour ces deux modes de communication.

L'analyse des travaux de la littérature montre que si plusieurs protocoles de sécurité ont été proposés, la quasi-totalité repose sur l'authentification comme étape primordiale et préalable pour garantir l'établissement de canaux de communication sécurisés entre les objets intelligents. Comme nous allons le voir dans la suite, ils diffèrent selon le type des objets communicants, les méthodes cryptographiques utilisées, les distances de communication et le domaine d'application.

2.6.1 Les protocoles d'authentification M2C

Dans le mode de communication M2C, un dispositif intelligent à faibles ressources peut avoir recourt à un ou à plusieurs proxys (mandataires) à proximité. Ces entités mandataires disposent de plus de ressources et peuvent ainsi effectuer les opérations coûteuses pour le compte des enti-

tés ayant moins de puissance de calcul ou échanger directement avec le cloud.

Dans [150], les auteurs proposent un protocole d'authentification utilisant des proxys et l'établissement d'une clé de session dans le contexte de l'assistance à la vie ambiante. Ils utilisent l'algorithme d'échange de clés DH (« Diffie-Hellman ») pour établir une authentification mutuelle entre le dispositif à faibles ressources d'un patient dans une maison intelligente et un serveur distant. La communication se fait via un cryptosystème à seuil (K, N) (threshold scheme) : l'émetteur divise le message en N différentes parties attribuées à différents proxys selon la formule de Lagrange [151] pour permettre aux destinataires recevant au moins K parties de calculer le message d'origine (le schéma résiste à la collision de $K-1$ proxys mais si les K proxys coopèrent ensemble, l'attaquant peut récupérer les informations originales).

Les auteurs dans [152] utilisent une technique de re-chiffrement par proxy PRE (« Proxy Re-Encryption ») pour faire un échange de clé authentifié afin de sécuriser les communications entre un objet intelligent et une autre entité plus puissante (par exemple, un serveur dans le cloud) ou entre deux objets intelligents, sans nécessiter le partage préalable d'une clé secrète entre entités source et destination. En substance, dans PRE, le proxy traduit les données chiffrées par une clé (la clé de l'initiateur) en données chiffrées par une autre clé (la clé du destinataire) sans pouvoir accéder aux données en clair.

Pourtant, Nuñez et al. [153] ont démontré que ce protocole n'assure pas la confidentialité persistante (« forward secrecy »), conduisant à des attaques d'usurpation d'identité et d'extension de longueur de clés.

Bien que les propositions reposant sur l'utilisation de proxy soient intéressantes, nos propositions de protocoles étant construites pour une connexion directe, il est plus pertinent d'examiner des travaux ayant plus de similarités.

Au contraire de ce qui est fait dans les propositions basées sur l'utilisation de proxys, dans la mesure où tous les calculs sont effectués par des objets intelligents limités en ressources, la plupart des autres propositions utilisent souvent des méthodes de chiffrement plus légères telles que la cryptographie sur courbes elliptiques.

Jin et al. [154] présentent un protocole d'authentification mutuelle pour améliorer la sécurité dans les systèmes d'administration de médicaments pour les patients dans les hôpitaux. Ce protocole fournit une authentification mutuelle entre les étiquettes RFID (bracelets RFID portés par les patients, emballages de médicaments étiquetés par des tags RFID) et le serveur (c'est-à-dire le cloud), en supposant que le canal de communication entre l'étiquette et le lecteur RFID n'est pas fiable (et en supposant par contre que le canal de communication entre le lecteur et le serveur est sécurisé). Pour sécuriser les futurs échanges sur le canal de communication RFID sans fil, une clé de session est construite lors du processus d'authentification entre les deux entités. Ce protocole

est décrit plus en détail dans la section 3.4.2.1.1 étant donné que l'une de nos contributions a été de proposer deux améliorations significatives.
Ces dernières années, plusieurs auteurs ont proposé des protocoles d'authentification mutuelle entre des tags RFID et des serveurs afin d'établir une clé de session dans le but de sécuriser efficacement les communications : par exemple, Zhao et al. [155] pour les environnements de soins de santé (healthcare), Alamr et al. [156], Liao et al. [157] et Dinarvand et al. [158] pour des environnements spécifiques. Lors de la phase de configuration (setup), le serveur génère pour chaque partie une paire de clés ECC publiques-privées et/ou un secret partagé qui seront utilisés lors de la phase d'authentification pour réaliser l'authentification mutuelle.

En dehors des protocoles d'authentification, il convient de noter qu'il existe une autre tendance visant à sécuriser la transmission de données depuis les dispositifs contraints vers des serveurs du cloud en utilisant des mécanismes de type chiffrement par attribut ABE (« Attribute Based Encryption »). Dans [159], Sharma et al. proposent une plate-forme de sécurité pour les données centralisées d'une infrastructure IoT au niveau de la couche réseau. Le but de cette approche est d'utiliser efficacement les données disponibles sur un serveur central. Ainsi, l'idée générale de leur système est de collecter les données à partir de plusieurs ressources (dispositifs médicaux, ordinateurs, etc.), de chiffrer ces données puis de les donner au serveur central. L'accès aux données sur le serveur pour les afficher ou exécuter des calculs dessus se fait sur la base d'un ensemble d'attributs. Pour présenter leur approche, les auteurs utilisent une application IoT réelle : un système e-health qui collecte les données et les enregistre dans le système centralisé. Ils combinent deux approches : une politique de chiffrement basée sur les attributs CPABE (« Ciphertext-Policy Attribute Based Encryption ») pour définir les règles d'accès aux données et le chiffrement fonctionnel FE (« Functional Encryption ») pour réaliser différentes fonctions sur les données chiffrées.
Dans [160], Li et al. proposent un schéma léger de partage de données pour le cloud mobile, LDSS (« Lightweight Data Sharing Scheme »), qui garantit un accès sécurisé aux données tout en respectant les contraintes qui s'imposent aux dispositifs mobiles en termes de cryptographie. Dans leurs travaux, ils ont adapté la technique CPABE utilisée dans le cloud classique à l'environnement du cloud mobile en modifiant de manière appropriée la structure de l'arborescence de contrôle d'accès et en externalisant, des dispositifs mobiles vers des serveurs proxy externes, une grande partie de la transformation de l'arborescence de contrôle d'accès CPABE nécessitant des calculs importants.

2.6.2 Les protocoles d'authentification M2M

Les systèmes M2M sont conçus pour fournir des services décentralisés via des dispositifs intelligents auto-organisés. Ils adoptent un modèle de communication distribué dans lequel deux nœuds quelconques peuvent établir une connexion où l'une des extrémités offre le service ou la ressource nécessaire à l'autre extrémité.

Dans ce mode de communication, comme pour le mode de communication M2C, l'authentification basée sur les proxys a été proposée.
Dans [161], Porambage et al. proposent un schéma similaire à [150] (présenté ci-dessus dans la section 2.6.1) dans lequel ils utilisent l'algorithme d'échange de clés DH pour établir une communication sécurisée de bout en bout E2E (« End-to-End ») entre les objets intelligents dans l'IoT.
Dans [162], un autre protocole d'authentification basé sur l'utilisation d'un proxy dans le contexte de la santé a été proposé par Amin et al. dans le but de préserver l'anonymat dans un protocole d'authentification mutuelle pour un réseau de capteurs médicaux sans fil WMSN (« Wireless Medical Sensor Network »). Dans l'architecture proposée, l'authentification est établie entre les dispositifs mobiles du personnel médical et des capteurs médicaux via le proxy. Néanmoins, dans [163], Jiang et al. ont revisité ce protocole qui est vulnérable aux attaques de vol de dispositifs mobiles, de désynchronisation et à l'exposition des clés de capteurs, afin de proposer un protocole d'authentification de bout en bout amélioré utilisant toujours un proxy.
Bien que ces propositions utilisant des proxys soient intéressantes, comme notre proposition de protocole M2M utilise une connexion directe, il est plus pertinent de se comparer avec des travaux plus proche. Malheureusement, après une analyse approfondie de la littérature et malgré l'étude des différents « surveys » récents sur la sécurité pour M2M [164] ou sur les protocoles d'authentification pour l'IoT [165] ou également sur l'IoT en général [166], article ayant pourtant une section dédiée pour M2M, nous n'avons identifié aucun protocole d'authentification M2M avec connexion directe.

2.7 Conclusion

Dans ce chapitre, nous avons donc présenté la sécurité dans la technologie IoT en général appliquée au domaine de la santé. En particulier, nous avons mis en évidence le besoin du respect de la vie privée des utilisateurs, nécessaire pour garantir l'adoption de l'IoT dans le secteur de la santé. De plus, ce chapitre inclut les attaques passives et actives auxquelles les systèmes IoT sont exposés. Dans ce contexte, le processus d'authentification est discuté comme étant la première barrière pour faire face à ces menaces de sécurité.
L'utilisation de mécanismes de sécurité adaptés aux types des communication M2M et M2C est

nécessaire pour l'utilisation adéquate des ressources que représente chaque technologie. Par conséquent, les cryptosystèmes basés sur les courbes elliptiques se posent en alternative efficace face à aux mécanismes de sécurité traditionnels comme l'incontournable RSA. En effet, le chiffrement sur les courbes elliptiques exploite un problème mathématique, qui est le problème de logarithme discret sur les courbes elliptiques. Ce qui permet au chiffrement ECC d'avoir une solidité égale à RSA pour des clés de longueur bien inférieure. Évidemment, nous n'avons présentés que les aspects généraux des algorithmes cryptographiques afin de ne pas sortir du cadre de ce travail.
Dans la suite de ce chapitre, nous avons analysé l'état de l'art des protocoles d'authentification existants en les classifiant selon les types de communication M2C et M2M. Aucun protocole d'authentification M2M avec connexion directe n'a été identifié.
Dans le chapitre suivant, les exigences de sécurité et fonctionnelles des protocoles d'authentification seront établis, puis, à partir de cela, trois protocoles d'authentification mutuelle seront proposés.

Chapitre 3

Les propositions de protocoles d'authentification M2C et M2M efficaces

Sommaire

3.1 Introduction

De nos jours, l'utilisation des dispositifs intelligents dans le domaine médical contribue entre autre à la surveillance médicale des patients en collectant, traitant et échangeant des données de santé selon les modes de communication M2C et M2M entre les personnes/objets. Néanmoins, ces dispositifs sont souvent utilisés dans des environnements ouverts et hostiles, et sont de ce fait exposés à diverses menaces. Pour cette raison, garantir des niveaux élevés de sécurité et de confidentialité devient crucial.

Pour contrer les menaces et les attaques auxquelles les communications de l'infrastructure IoT sont exposées, un ensemble de services de sécurité doit être garanti. La première ligne de défense est la mise en place d'un service d'authentification pour garantir que les entités communiquantes sont bien qui elles prétendent être par le biais de protocoles d'authentification. Dans le contexte de l'IoT, au vu des ressources contraintes des dispositifs connectés, les protocoles d'authentification reposent souvent sur des algorithmes de chiffrement plus légers et adaptés aux faibles ressources disponibles, comme ceux basés sur les courbes elliptiques. La phase d'authentification est souvent une étape préalable obligatoire pour permettre aux différentes parties de communiquer ensuite en toute sécurité. L'objectif de ce chapitre est donc de proposer des protocoles d'authentification mutuelle M2C et M2M efficaces pouvant être utilisés dans diverses applications de santé basées sur l'IoT.

Nos principales contributions mises en œuvre dans ce chapitre peuvent se résumer ainsi :

1. L'ajout de deux améliorations (sur les deux côtés de la communication) sur un protocole d'authentification mutuelle M2C récemment proposé pour un système de santé basé sur la technologie RFID.
2. La proposition d'un nouveau protocole d'authentification mutuelle M2C plus efficace et scalable qui se base principalement sur l'utilisation de fonctions de hachage.
3. La proposition d'un nouveau protocole d'authentification mutuelle M2M utilisant les ECC.
4. La validation des protocoles proposés à l'aide de deux outils de vérification formelle, AVISPA et ProVerif, et l'évaluation des performances sur des dispositifs IoT.

3.2 Les exigences de sécurité

Les canaux de communication entre deux objets intelligents (M2M) ou entre un objet intelligent et un serveur distant dans le cloud (M2C) étant sujets à diverses attaques et les applications médicales exigeant que les données relatives à un patient soient gérées de manière à garantir leur

confidentialité et leur intégrité tout en respectant la vie privée, l'authentification est une étape primordiale obligatoire pour permettre l'établissement des clés de session nécessaires à la sécurisation des futurs échanges de données entre les différentes entités des systèmes M2C et M2M.

Les exigences de sécurité (ES) souhaitées sont donc :

(ES1) Authentification : Deux entités communicantes en mode M2M ou en mode M2C doivent s'authentifier mutuellement au préalable et avant toute autre interaction ultérieure. Chaque entité devrait donc fournir une preuve de son identité.

(ES2) Confidentialité, intégrité et respect de la vie privée : Pour garantir la confidentialité, l'intégrité des données échangées et le respect de la vie privée des entités communicantes, une nouvelle clé de session est requise à chaque authentification.

3.3 Les exigences fonctionnelles

En plus des exigences de sécurité, les exigences fonctionnelles (EF) souhaitées sont :

(EF1) Efficacité :

- (EF1a) Temps de traitement et scalabilité de l'authentification : Dans le mode de communication M2C, étant donné que les serveurs dans le cloud ont besoin de conserver des informations relatives à un grand nombre d'objets intelligents dans leurs bases de données, il est nécessaire que l'authentification s'effectue aussi rapidement que possible et dans un temps qui soit indépendant du nombre d'objets intelligents dans le système. De manière similaire, dans le mode de communication M2M, il est également nécessaire de disposer d'un temps d'authentification non lié au nombre d'objets intelligents dans le système.
- (EF1b) Minimiser le nombre de messages échangés : Afin d'assurer l'efficacité des communications pendant la phase d'authentification mutuelle, le nombre de messages et la taille des données échangées doivent être minimisés.

(EF2) Légèreté : Afin de respecter les contraintes de ressources des objets intelligents, le protocole d'authentification doit être léger.

- (EF2a) Calcul : Les primitives de chiffrement utilisées sur les objets intelligents doivent avoir un coût de calcul inférieur aux algorithmes cryptographiques traditionnels (par exemple RSA, AES)
- (FF2b) Stockage : Le stockage des données sur les objets intelligents doit être minimisé.

(EF3) Résilience : Pour protéger le système contre les attaques de déni de service et de désynchronisation, les protocoles d'authentification doivent être conçu de façon « sans état » (« stateless » en anglais) [167] : c'est-à-dire que les entités communicantes n'ont pas besoin de

conserver les informations de session pour assurer une bonne exécution du processus d'authentification.

Pour le mode de communication M2M, une exigence fonctionnelle supplémentaire s'ajoute :

(EF4) Une solution distribuée : Les objets intelligents doivent être capables de communiquer directement sans nécessiter l'intervention en ligne d'un serveur central au moment de la phase d'authentification.

3.4 Les propositions de protocoles d'authentification M2C et M2M

Avant de détailler nos deux protocoles d'authentification M2C et le protocole d'authentification M2M, nous donnons dans le Tableau 3.1 les notations utilisées dans leurs descriptions. Puis, nous décrivons le modèle d'attaques considéré et les outils de vérification automatique utilisés pour la validation de ces protocoles. La première proposition de protocole d'authentification M2C étant une version améliorée de celle de Jin et al. [154] citée précédemment dans la section 2.6.1, par souci de cohérence, elle est présentée avec les mêmes notations que celles utilisées pour d'autres protocoles en section 3.4.2.1.1. Ensuite, pour chaque proposition de protocole, une analyse informelle et une validation à l'aide de deux outils de vérification formelle seront effectuées. Il est à noter que les protocoles d'authentification proposés ont une connexion directe (c'est-à-dire sans proxy) et que tous nos protocoles d'authentification mutuelle ont trois étapes (échanges) et comportent deux phases :

- Une phase de configuration : Cette première phase peut se dérouler en mode hors ligne et elle s'effectue normalement une seule fois dans le cycle de vie des objets intelligents (par exemple lors de leur initialisation ou de leur personnalisation). Elle consiste essentiellement à fournir les accréditations pour tous les protocoles présentés.
- Une phase d'authentification : Cette seconde phase consiste généralement en trois échanges dans le but de réaliser l'authentification mutuelle des entités légitimes et elle est effectuée à chaque fois qu'une authentification est requise.

3.4.1 Le modèle de l'attaquant et les outils de vérification formelle

Le modèle de l'attaquant considéré pour les protocoles proposés est le modèle largement utilisé de Dolev-Yao [115]. Ce modèle suppose que l'attaquant a le contrôle complet de tout le réseau de communication. De ce fait, l'attaquant peut surcharger, intercepter et synthétiser n'importe quel message transmis sur un réseau. L'attaquant est considéré aussi comme pouvant être n'importe quel agent parmi ceux impliqués dans l'échange à travers le protocole. Néanmoins, il n'est pas en mesure de casser les primitives sécurisées du chiffrement qui sont utilisées dans la

TABLEAU 3.1 – Notations utilisées dans la description des protocoles

Notation	**Signification**
n	Nombre d'objets intelligents dans le système
S	Désigne un Serveur
SO	Désigne un objet intelligent de façon générique
T_i	Désigne un objet intelligent particulier T_i où $i \in [1,n]$
q, p	Deux grand nombres premiers
$\mathbb{Z}_p^*$	Un corps fini
E	Une courbe elliptique définie sur un corps fini $\mathbb{F}_q$ par l'équation $y^2 = x^3 + ax + b$, où $a, b \in \mathbb{F}_q$
P	Un générateur de E d'ordre p
(q, a, b, p, P)	Désigne les paramètres du domaine définissant E
N	Désigne la taille en bits d'une clé privée en utilisant la courbe E ; $N = \log_2(p)$
H	Désigne une fonction de hachage sécurisée et résistante aux collisions
H_j	Désigne une fonction de hachage sécurisée et résistante aux collisions (utilisée uniquement dans Jin et al.'s protocol)
$r \xleftarrow{\$} \mathbb{Z}_p^*$	Désigne que r est un nombre aléatoire $\in \mathbb{Z}_p^*$
$A + B$	Selon le contexte, désigne l'addition de point entre deux points A et B ou l'addition scalaire entre deux scalaires A et B
$A \cdot B$	Selon le contexte, désigne la multiplication de point entre un scalaire A et un point B ou la multiplication scalaire entre deux scalaires A et B
$A \oplus B$	Désigne l'opération XOR entre les valeurs de A et B
$A \| \| B$	Désigne l'opération de concaténation de A et de B
$A \stackrel{?}{=} B$	Désigne la vérification de l'égalité entre A et B
$A = B$	Désigne que la variable A reçoit la valeur de l'expression B
Id_{T_i}	Identité de T_i
HId_{T_i}	Clé de l'entrée de hachage liée à l'identité de T_i dans la base de données
x_S, P_S	Désigne les clés privée et publique du serveur S, où $P_S = x_S \cdot P$
x_i, P_i	Désigne les clés privée et publique de T_i, où $P_i = x_i \cdot P$
$\sigma_i = (\bar{P}_i, z_i)$	Désigne la signature de Schnorr [145] de la clé publique P_i de T_i par le serveur. Le serveur choisit une valeur aléatoire $s_i \in \mathbb{Z}_p^*$ et calcule $\bar{P}_i = s_i \cdot P$ et $z_i = s_i + H(P_i, \bar{P}_i) \cdot x_S$. La signature résultante est σ_i

construction du protocole c'est-à-dire il lui est impossible de déchiffrer un message chiffré ou d'obtenir des informations sur le message en clair à partir du message chiffré sans connaître la clé de déchiffrement/chiffrement utilisée.

Pour valider formellement les protocoles d'authentification proposés, les outils de vérification AVISPA (présenté en Annexe A) [168] et ProVerif (présenté en Annexe B) [169] ont été utilisés. Les deux utilisent le modèle de l'attaquant Dolev-Yao.

3.4.2 La description des protocoles d'authentification M2C

Dans cette section, deux protocoles d'authentification directs M2C sont proposés et validés. Le premier consiste en une amélioration du protocole d'authentification de Jin et al. [154]. Le second protocole qui se base principalement sur l'utilisation de fonctions de hachage, est proposé car il réduit à la fois l'usage des ressources de stockage et de calcul sur les objets intelligents, mais aussi celles de calcul sur le serveur. De plus, ce dernier offre plus de scalabilité et de flexibilité que le premier protocole puisqu'il permet l'authentification mutuelle d'objets intelligents et de multiples serveurs, à condition que ces derniers connaissent les identités des objets intelligents et les clés associées de leurs entrées de hachage dans la base de données.

3.4.2.1 Améliorations du protocole de Jin et al.

Avant d'expliquer les améliorations apportées au protocole de Jin et al., la section suivante présente sa version originale.

3.4.2.1.1 La version originale du protocole de Jin et al.

Le protocole d'authentification de Jin et al [154] est basé sur l'utilisation d'ECC et est conçu pour les dispositifs à ressources limitées, tels que les tags RFID. Ce protocole peut être classé comme un protocole d'authentification M2C étant donné que le serveur peut être hébergé dans le cloud mais aussi que les tags RFID peuvent être considérés comme étant des objets intelligents.

Phase de configuration

Pour commencer, le serveur S génère et partage avec les autres entités les paramètres du système (q, a, b, p, P) à utiliser pour la génération de la courbe elliptique, E. Par la suite, le serveur choisit une valeur aléatoire $x_S \in \mathbb{Z}_p^*$ comme étant sa clé privée et calcule sa clé publique comme suit : $P_S = x_S \cdot P$. Ensuite, chaque objet intelligent T_i reçoit un identifiant unique (nommé X_T dans l'article d'origine et nommé ici Id_{T_i}) envoyé par le serveur, qui est un point sur E, et la clé publique de ce dernier P_S.

Phase d'authentication

Cette phase est composée par les trois échanges représentés dans la Figure 3.1.

1. Le serveur S choisit une valeur aléatoire $r_S \in \mathbb{Z}_p^*$ et calcule $R_S = r_S \cdot P$. Ensuite, S envoie (R_S) à l'objet intelligent T_i.

2. À la réception, l'objet intelligent T_i génère d'abord une valeur aléatoire $r_i \in \mathbb{Z}_p^*$ puis calcule $R_i = r_i \cdot P$ et $TK_i = r_i \cdot P_S$. Ce dernier est utilisé pour masquer Id_{T_i} dans $Auth_i = Id_{T_i} \oplus H_1(R_S, TK_i)$. Ensuite, il envoie $(R_i, Auth_i)$ au serveur.

3. Le serveur S calcule $TK_S = x_S \cdot R_i$ et $Id_{T_i} = Auth_i \oplus H_1(R_S, TK_S)$. Ensuite, il recherche Id_{T_i} dans sa base de données. Si le serveur ne trouve pas Id_{T_i}, à savoir que l'entrée n'existe pas dans sa base de données, il arrête immédiatement la session.

   Dans le cas contraire, le serveur calcule $e = H_2(R_S, R_i, Id_{T_i})$ et $Auth_S = (x_S \cdot e + r_S)$. Puis, il envoie $Auth_S$ à l'objet intelligent T_i.

4. L'objet intelligent T_i calcule $e = H_2(R_S, R_i, Id_{T_i})$ et compare $Auth_S \cdot P$ à $e \cdot P_S + R_S$. Si les valeurs ne sont pas équivalentes, la session est arrêtée. Sinon, le serveur est authentifié par l'objet intelligent et l'authentification mutuelle est réalisée.

À la fin de la phase d'authentification, le serveur S et l'objet intelligent T_i partagent une clé de session e qui peut être utilisée pour assurer la confidentialité, l'intégrité et le respect de la vie privée pendant les échanges ultérieurs.

Analyse de sécurité

Étant donné que le protocole est sécurisé d'après Jin et al., son analyse de sécurité ne sera pas effectuée. Cependant, le lecteur peut se référer à l'analyse de sécurité de la version améliorée qui elle sera présentée. Néanmoins, il faut relever que dans ce protocole, d'une certaine manière, le serveur S authentifie implicitement l'objet intelligent T_i en fonction de l'existence de Id_{T_i} dans la base de données. Ce n'est pas un gros problème puisqu'un attaquant ne devrait pas être capable de calculer la clé partagée e utilisée pour des échanges ultérieurs. Toutefois, ce problème n'existera plus dans l'amélioration que nous proposerons. Un autre point à mentionner est que Jin et al. n'utilisent pas la propriété que Id_{T_i} soit un point de E. Cette valeur pourrait d'ailleurs être n'importe quelle chaîne ayant le même nombre de bits. En fait, c'est même plutôt une contrainte pour eux puisqu'ils ont alors besoin de deux fonctions de hachage distinctes, une dont la taille de la valeur de la sortie est celle du point de la courbe (utilisé pour masquer $Auth_i$ avec un xor) et d'une seconde dont la sortie est un scalaire (utilisé pour calculer $Auth_S$ lors de l'opération modulaire).

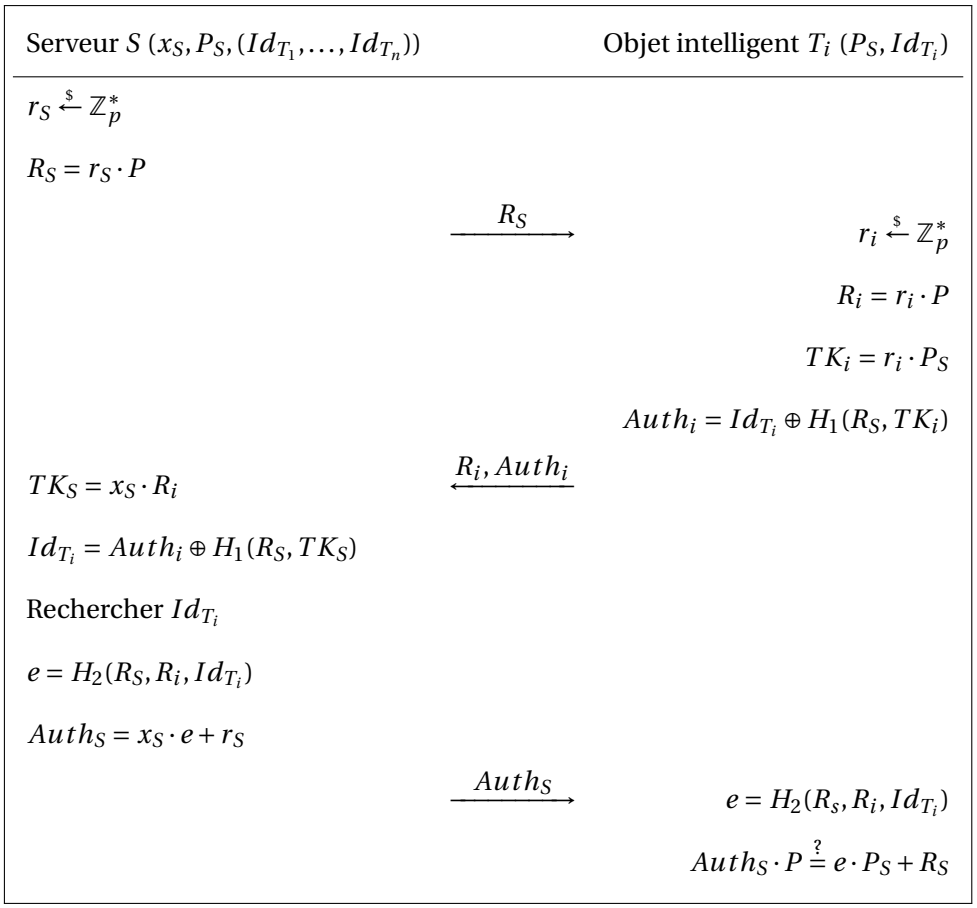


FIGURE 3.1 – Phase d'authentification du protocole M2C de Jin et al.

3.4.2.1.2 *Les deux améliorations principales : réduction de la complexité de la recherche du côté du serveur et du calcul effectué par l'objet intelligent*

Dans le protocole original présenté ci-dessus, pour authentifier implicitement un objet intelligent T_i, le serveur S stocke tous les identifiants des objets intelligents dans sa base de données. De ce fait, après la réception de $Auth_i$, il calcule l'identifiant de l'objet intelligent, Id_{T_i} et vérifie si l'objet intelligent T_i existe dans sa base de données (l'étape Rechercher Id_{T_i}, comme illustré dans la Figure 3.1). En termes de complexité, cela signifie que pour n objets intelligents dans le système, cette recherche peut nécessiter jusqu'à n accès à la base de données du serveur si l'identifiant est stocké à la dernière position. Pour réduire cette complexité linéaire de $O(n)$ à une complexité constante (c'est-à-dire $O(1)$), en s'inspirant du protocole de Bonnefoi et al. [170] (qui propose un ensemble de protocoles d'authentification extrêmement légers entre une étiquette RFID et une base de données dont l'accès est en $O(1)$ pour maintenir le coût de calcul du côté de la base de données lors de l'authentification des étiquettes à une complexité constante), la première amélioration consiste à modifier le protocole en ajoutant la valeur HId_{T_i} qui est le résultat de la fonction de hachage H appliquée à l'identifiant Id_{T_i} dans la mémoire de l'objet intelligent T_i lors de son en-

registrement dans le système à la phase de configuration. Cette valeur représente la clé de l'entrée de la table de hachage de la base de données, ce qui permet d'avoir un accès direct à l'identifiant Id_{T_i} associé. La consommation mémoire de l'objet intelligent reste inchangé malgré cet ajout car la taille individuelle de chaque valeur Id_{T_i} et HId_{T_i} est choisie de sorte qu'elle soit de la même taille que la moitié d'un point de E. Aussi, avec l'utilisation de ces deux valeurs, le protocole proposé s'exécute plus rapidement du côté du serveur, en plus de disposer d'une authentification explicite (voir l'analyse de sécurité ci-dessous). En outre, il est à noter que la taille du message 2 (R_i, $Auth_i$, M) est identique à celle du message 2 du protocole de Jin et al.

La deuxième amélioration consiste à alléger les calculs complexes que les objets intelligents doivent réaliser en modifiant le message $Auth_S$ envoyé par S. S effectue maintenant la multiplication scalaire que l'objet intelligent T_i avait à réaliser pour authentifier le serveur. Une multiplication scalaire est donc économisée sur l'objet intelligent. En contre-partie, la taille du message $Auth_S$ est augmentée par rapport à Jin et al.

Une troisième amélioration mineure consiste à remplacer les deux fonctions de hachage utilisées dans la version originale par une seule car cela ne contribue pas à la sécurité du protocole mais au contraire consomme plus d'espace mémoire ou d'espace sur le circuit intégré ICC (« Integrated Chip Circuit ») si elles sont implémentées dans le hardware.

La version améliorée du protocole d'authentification est décrite à la Figure 3.2.

Phase de configuration

Comme pour le déroulement de la phase de configuration du protocole original, le serveur génère et partage avec les autres entités les paramètres du système (q, a, b, p, P) à utiliser pour la courbe elliptique E. Ensuite, il choisit une valeur aléatoire $x_S \in \mathbb{Z}_p^*$ comme étant sa clé privée et calcule sa clé publique sous la forme de $P_S = x_S \cdot P$. Ensuite, pour chaque objet intelligent T_i, S génère un identifiant aléatoire Id_{T_i} qu'il lui envoie avec la clé de hachage (de la base de données permettant de stocker les identifiants des objets intelligents) qui lui correspond $HId_{T_i} = H(Id_{T_i})$ et avec la clé publique du serveur P_S. Il convient de mentionner que si T_i présente une contrainte d'espace mémoire, il peut éviter de stocker HId_{T_i} et le recalculer à chaque demande d'authentification mais cela ralentit le processus d'authentification.

Phase d'authentification

Dans ce qui suit, les étapes et interactions entre le serveur S et un objet intelligent spécifique T_i sont décrites :

1. Le serveur S choisit une valeur aléatoire $r_S \in \mathbb{Z}_p^*$ et calcule $R_S = r_S \cdot P$. Puis, il envoie R_S à T_i.

Serveur S $(x_S, P_S, ((Id_{T_1}, HId_{T_1}), \ldots, (Id_{T_n}, HId_{T_n}))$		Objet intelligent T_i $(P_S, (Id_{T_i}, HId_{T_i}))$
$r_S \xleftarrow{\$} \mathbb{Z}_p^*$		
$R_S = r_S \cdot P$		
	$\xrightarrow{R_S}$	$r_i \xleftarrow{\$} \mathbb{Z}_p^*$
		$R_i = r_i \cdot P$
		$Mask = r_i \cdot P_S$
		$Auth_i = Id_T \oplus H(R_S, Mask)$
		$M = HId_{T_i} \oplus H(Mask)$
	$\xleftarrow{R_i, Auth_i, M}$	
$Mask = x_S \cdot R_i$		
$HId'_{T_i} = M \oplus H(Mask)$		
Recherche de $HId'_{T_i} \rightarrow Id'_{T_i}$		
$Id_{T_i} = Auth_i \oplus H(R_S, Mask)$		
$Id'_{T_i} \stackrel{?}{=} Id_{T_i}$		
$e = H(R_S, R_i, Id_{T_i})$		
$Auth_S = (x_S \cdot e + r_S) \cdot P$		
	$\xrightarrow{Auth_S}$	$e = H(R_S, R_i, Id_{T_i})$
		$Auth_S \stackrel{?}{=} e \cdot P_S + R_S$

FIGURE 3.2 – Amélioration du protocole d'authentification M2C de Jin et al.

2. À la réception, l'objet intelligent T_i génère d'abord une valeur aléatoire $r_i \in \mathbb{Z}_p^*$ puis calcule $R_i = r_i \cdot P$ et $Mask = r_i \cdot P_S$ qui est utilisé pour masquer Id_{T_i} et HId_{T_i} dans $Auth_i = Id_{T_i} \oplus H(R_S, Mask)$ et $M = HId_{T_i} \oplus H(Mask)$. Ensuite, il envoie $(R_i, Auth_i, M)$ au serveur.

3. Le serveur S calcule $Mask = x_S \cdot R_i$ utilisé pour rechercher $HId'_{T_i} = M \oplus H(Mask)$. Ensuite, il utilise HId'_{T_i} pour rechercher Id'_{T_i} dans sa base de données.

   Si le serveur ne trouve pas Id'_{T_i}, à savoir que l'entrée n'existe pas dans sa base de données, il n'arrête pas immédiatement la session pour ne pas divulguer d'informations à un attaquant potentiel. Il arrête donc les sessions de communication après un délai déterminé (en d'autres termes, il doit être programmé pour répondre en temps constant aux trois opérations de recherche de Id'_{T_i}, de calcul de Id_{T_i} et de comparaison de Id'_{T_i}). Sinon, il calcule $Id_{T_i} = Auth_i \oplus H(R_S, Mask)$ puis il compare Id_{T_i} à Id'_{T_i}. Si les valeurs ne sont pas équivalentes, le serveur arrête la session. Sinon, le dispositif est authentifié.

Ensuite, le serveur calcule $e = H(R_S, R_i, Id_{T_i})$ et $Auth_S = (x_S \cdot e + r_S) \cdot P$. Puis, il envoie $Auth_S$ à l'objet intelligent T_i.

4. L'objet intelligent T_i calcule $e = H(R_S, R_i, Id_{T_i})$ et compare $Auth_S$ reçu à $e \cdot P_S + R_S$. Si les valeurs ne sont pas égales, la session est arrêtée. Sinon, le serveur est authentifié par l'objet intelligent et l'authentification mutuelle est réalisée.

Comme pour la version originale, à la fin de la phase d'authentification, le serveur S et l'objet intelligent T_i partagent une clé de session e qui peut être utilisée pour garantir la confidentialité, l'intégrité et le respect de la vie privée pendant les échanges ultérieurs.

Avant de présenter l'analyse de sécurité de ce protocole, il convient de noter que certaines optimisations peuvent être effectuées entre les messages 2 et 3 du côté de l'objet intelligent (c'est-à-dire lorsque le serveur S est en train de vérifier la légitimité de l'objet intelligent). Le fait que T_i peut précalculer $e = H(R_S, R_i, Id_{T_i})$ et $e \cdot P_S + R_S$ contribue à accélérer l'authentification de S à la réception du message 3, c'est-à-dire $Auth_S$ puisqu'il effectue qu'une seule comparaison. Cette astuce peut également être utilisée dans le protocole de Jin et al., mais cela n'a pas été mentionné. À vrai dire, ce n'est que récemment, dans le domaine de la technologie RFID, le domaine pour lequel le protocole de Jin et al. a été développé, que les concepteurs de protocole ont utilisé le fait que l'étiquette RFID est alimentée dès lors qu'elle est présente dans le champs du lecteur pour effectuer des pré-calculs [171].

Analyse de sécurité

Dans ce qui suit, le protocole est analysé en ce qui concerne les exigences de sécurité et certaines propriétés supplémentaires.

- Authentification mutuelle :
  - — Authentification de l'objet intelligent : L'adversaire ne peut pas générer un message légitime $(R_i, Auth_i, M)$ d'un objet intelligent T_i, puisqu'il n'est pas en mesure d'obtenir la paire (identifiant Id_{T_i}, la clé associée HId_{T_i}) de l'objet intelligent, ainsi que la valeur aléatoire r_i qui sont utilisées pour calculer $Auth_i$ et M. Cependant, le serveur S peut calculer $Mask$ (puisque $x_S \cdot R_i$ est égal à $r_i \cdot P_S$) pour obtenir HId'_{T_i} et vérifier si l'entrée est existante. De plus, dans le cas où l'entrée n'existe pas, le serveur S ne doit pas arrêter immédiatement la session ; il doit répondre dans un délai déterminé pour éviter toute divulgation d'informations à un attaquant. Afin de rester au même niveau de sécurité que le protocole de Jin et al., nous avons décidé de ne pas augmenter le volume de données à stocker sur l'objet intelligent dans cette version améliorée (les tailles de Id_{T_i} et HId_{T_i} sont choisies pour occuper le même espace que Id_{T_i} dans Jin et al.). Si l'entrée n'existe pas, S doit attendre de la même façon que si la correspondance entre Id'_{T_i} et

Id_{T_i} existait vraiment. Dans le cas où l'entrée correspond, S compare le Id'_{T_i} reçu à Id_{T_i}. S'ils sont différents, S arrête la session. Sinon, l'objet intelligent T_i est authentifié.

— Authentification du serveur : L'adversaire ne peut pas générer un message légitime $Auth_S$ de S, puisqu'il n'est pas en mesure d'obtenir e résultant de l'utilisation de l'identifiant Id_{T_i} de l'objet intelligent T_i. En effet, la sécurité de l'authentification du serveur repose sur e puisque P_S est connu et qu'un adversaire passif peut bien obtenir R_S. À la réception de $Auth_S$, l'objet intelligent T_i doit calculer e pour ensuite vérifier si la valeur de $Auth_S$ reçue correspond à celle calculée par $e \cdot P_S + R_S$. Si une égalité est trouvée alors l'objet intelligent authentifie S. Il est intéressant de constater que le déplacement de la multiplication scalaire $(e \cdot P_S + R_S) \cdot P$ depuis l'objet intelligent vers le serveur (c'est-à-dire dans $Auth_S$), dans le but de renforcer l'authentification, repose uniquement sur la possibilité de calculer e, au lieu d'y ajouter aussi la connaissance de x_S comme dans le protocole de Jin et al. C'est notre choix et il revient aux développeurs du protocole de décider d'utiliser ou non cette amélioration, même si le niveau de sécurité est identique dans les deux cas.

- Confidentialité, intégrité et respect de la vie privée : À la fin du protocole, les deux parties S et T_i, partagent une nouvelle clé de session e, à laquelle ils ont contribué. En effet, elle repose sur des valeurs aléatoires r_S, r_i générées pour chaque session et sur Id_{T_i}. De ce fait, S et T_i peuvent l'utiliser pour dériver des clés permettant de garantir la confidentialité, l'intégrité et le respect de la vie privée pendant les échanges ultérieurs.
- Disponibilité :
  — Résistance aux attaques de désynchronisation : Dans le protocole proposé, étant donné qu'il ne contient pas d'étape de mise à jour de clé et qu'il a une construction de type « stateless » (sans état), les clés ne peuvent pas être désynchronisées et le protocole peut être repris depuis le début à n'importe quelle étape de l'authentification.
  — Résistance aux attaques DoS : Le protocole proposé étant sans état, chaque transaction est considérée comme étant nouvelle. Du côté des objets intelligents, une nouvelle transaction d'authentification ne consomme pas plus d'espace mémoire que la précédente. Certes, le lancement d'une authentification utilise des ressources de calcul mais c'est inévitable. Le cas le plus défavorable du côté des objets intelligents dépend de l'implémentation et du type de la mémoire utilisée : si l'objet intelligent utilise une technologie de mémoire non volatile comme la Flash ou l'EEPROM pour stocker ses calculs intermédiaires, les cellules mémoire peuvent mourir après 10^5 cycles d'écriture. Cependant, s'il utilise de la mémoire FRAM, elle durera jusqu'à 10^{12} cycles et s'il utilise de la RAM seulement, le nombre de cycles sera infini. Du côté du serveur, puisque

le serveur doit être capable de gérer plusieurs authentifications d'objets intelligents en même temps, chaque nouvelle transaction d'authentification consomme certaines ressources mémoire. Cependant, dans le cas d'une attaque par déni de service, le mécanisme de recherche efficace proposé limite grandement son impact, puisque le serveur ne passe pas son temps à faire des calculs et à interroger la base de données pour un faux HId'_{T_i}.

- Scalabilité : Grâce à l'efficacité du mécanisme de recherche proposé, le serveur S calcule d'abord le masque ($Mask$), et ensuite il vérifie l'existence de HId'_{T_i} dans sa base de données et éventuellement il vérifie la correspondance de Id'_{T_i} et Id_{T_i} en temps constant, c'est-à-dire en $O(1)$.
- Anonymat de l'objet intelligent : Dans le protocole proposé, l'identifiant de l'objet intelligent et la clé de hachache associée (Id_{T_i}, HId_{T_i}) sont masqués par une fonction utilisant $Mask$. Aussi, bien qu'un adversaire passif puisse obtenir R_S et que P_S soit publique, il ne peut pas calculer $Mask$ sans la valeur aléatoire r_i et ne peut donc pas calculer Id_{T_i} ou HId_{T_i}. De plus, dans chaque nouvelle session, le serveur S et l'objet intelligent T_i génèrent de nouvelles valeurs aléatoires distinctes r_S et r_i. Par conséquent, l'adversaire ne peut pas localiser l'emplacement de l'objet intelligent et donc notre protocole assure l'anonymat.

⋆ Analyse formelle avec AVISPA ⋆

Comme le détail de la spécification est donné en annexe, nous ne présentons dans cette section que les points importants et les résultats.

▷ Spécification du protocole

Le script HLPSL de notre protocole est présenté dans l'Annexe A.2.

Il existe deux rôles de base S et T qui sont relatif à l'activité du $Serveur$ et de T_i. Le concept fondamental du protocole consiste à conserver les valeurs Id_{T_i}, HId_{T_i}, x_S, r_S et r_i `secret` tout au long de la phase d'authentification du protocole entre S et T. L'authentification mutuelle est réalisée via les objectifs `witness` et `request`. Les détails sont fournis en annexe.

▷ Résultats de la vérification

Les résultats après l'exécution de notre protocole codé en HLPSL sont donnés dans le Tableau 3.2.

Comme le montre le Tableau 3.2, AVISPA retourne SAFE à partir de ses deux back-ends On-the-Fly Model-Checker (OFMC) et CL-based Model-Checker (CL-AtSe). Par contre, SAT-based Model-Checker (SATMC) et Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) retournent INCONCLUSIVE en raison d'opérations non supportées. Ce qui

TABLEAU 3.2 – Résultats de validation AVISPA

Moteur AVISPA	Résultat
$OFMC$	SAFE
$CL-AtSe$	SAFE
$SATMC$	INCONCLUSIVE
$TA4SP$	INCONCLUSIVE

signifie que AVISPA ne peut pas trouver d'attaques contre notre protocole.

⋆ Analyse formelle avec ProVerif ⋆

Dans les paragraphes suivants, les points importants de la spécification du protocole et les résultats sont mis en avant.

▷ Spécification du protocole

Les scripts de Pi Calculus appliqués sont donnés dans l'Annexe B.2. Comme pour AVISPA, il existe deux rôles de base `SERVERS` et `SOI` qui expliquent l'activité de $Server$ et de T_i. Les valeurs secrètes sont vérifiées à l'aide de requêtes (« queries »). L'authentification mutuelle entre T_i et $Server$ est modélisée à l'aide d'événements mappés dans les sous-processus et les requêtes de `SOI` et de `SERVERS`. Les détails sont fournis en annexe.

▷ Résultats de la vérification

Les résultats sont disponibles dans l'Annexe B.3. Ils montrent que le confidentialité (« secrecy ») de `idt`, `hidt`, `xs`, `ri`, `rs` est préservée par le protocole et que l'authentification mutuelle entre `SERVERS` et `SOI` est réalisée.

3.4.2.2 Protocole d'authentification basé sur l'utilisation de fonctions de hachage

Dans cette section, nous présentons un autre protocole d'authentification qui se base également sur les courbes elliptiques mais axé davantage sur l'utilisation de fonctions de hachage. Le principal avantage de cette solution par rapport à la précédente amélioration de Jin et al. [154] est qu'elle est plus scalable et flexible. En effet, ce protocole permet l'authentification mutuelle entre des objets intelligents et différents serveurs, à condition que ces derniers possèdent les identités des objets intelligents. Autrement dit, si un serveur tombe en panne, l'authentification peut s'effectuer auprès d'un autre serveur, sous réserve que ce dernier connaisse l'identité de l'objet intelligent et la clé de hachage de l'entrée associée.

Ce protocole comprend deux phases, à savoir la phase de configuration et la phase d'authentification.

Serveur S $(Id_{T_1}, HId_{T_1}), \ldots, (Id_{T_n}, HId_{T_n})$		Objet intelligent T_i (Id_{T_i}, HId_{T_i})
$r_S \xleftarrow{\$} \mathbb{Z}_p^*$		
$R_S = r_S \cdot P$		
	$\xrightarrow{R_S}$	$r_i \xleftarrow{\$} \mathbb{Z}_p^*$
		$R_i = r_i \cdot P$
		$K = r_i \cdot R_S$
		$Auth_i = H(Id_{T_i} \| 2 \| K)$
	$\xleftarrow{R_i, Auth_i, M}$	$M = HId_{T_i} \oplus K$
$K = r_S \cdot R_i$		
$HId'_{T_i} = M \oplus K$		
Recherche de $HId'_{T_i} \rightarrow Id'_{T_i}$		
$Auth'_i = H(Id'_{T_i} \| 2 \| K)$		
$Auth'_i \stackrel{?}{=} Auth_i$		
$Auth_S = H(Id_{T_i} \| 1 \| K)$		
	$\xrightarrow{Auth_S}$	$Auth'_S = H(Id_{T_i} \| 1 \| K)$
		$Auth'_S \stackrel{?}{=} Auth_S$

FIGURE 3.3 – Protocole d'authentification basé sur l'utilisation de fonctions de hachage

Phase de configuration

Durant cette phase, le serveur génère et partage avec les autres entités les paramètres du système (q, a, b, p, P) à utiliser pour la courbe elliptique E. À ce stade, pour chaque objet intelligent T_i, S génère un Id_{T_i} aléatoire comme étant l'identifiant de l'objet intelligent et le lui envoie avec $HId_{T_i} = H(Id_{T_i})$ qui est la clé de la base de données (table de hachage utilisée pour stocker les identifiants des objets intelligents). En cas de contrainte en espace mémoire sur l'objet T_i, ce dernier peut calculer HId_{T_i} lui-même à chaque demande d'authentification au lieu de le stocker en mémoire. Néanmoins, cela risque de ralentir le processus d'authentification.

Phase d'authentification

Cette phase est illustrée par la Figure 3.3.

1. Le serveur S choisit une valeur aléatoire $r_S \in \mathbb{Z}_p^*$ et calcule $R_S = r_S \cdot P$. Puis, il envoie R_S à T_i.
2. À la réception, T_i génère d'abord une valeur aléatoire $r_i \in \mathbb{Z}_p^*$, puis calcule $R_i = r_i \cdot P$ et $K = r_i \cdot R_S$. Ce dernier est utilisé pour masquer HId_{T_i} dans $M = HId_{T_i} \oplus K$ (où seule la coordonnée x du point K est utilisée pour le masque; nous pourrions utiliser $H(K)$ mais cela nécessiterait un appel supplémentaire à la fonction de hachage). Il calcule également $Auth_i = H(Id_{T_i} \| 2 \| K)$, puis envoie $(R_i, Auth_i, M)$ au serveur.
3. Le serveur S calcule d'abord $K = r_S \cdot R_i$ et l'utilise pour trouver $HId'_{T_i} = M \oplus K$. Ensuite, il recherche Id'_{T_i} dans sa base de données à l'aide de HId'_{T_i}.

   Si Id'_{T_i} n'est pas trouvé, c'est-à-dire que l'entrée n'existe pas dans sa base de données, le serveur S n'interrompt pas immédiatement la session afin d'éviter de divulguer des informations à un attaquant potentiel. Donc, il arrête la session après un certain délai (S doit être programmé pour répondre dans un délai correspondant aux trois opérations suivantes : rechercher Id'_{T_i}, calculer $Auth'_i$ et effectuer la comparaison avec $Auth_i$).

   Sinon, il calcule $Auth'_i = H(Id'_{T_i} \| 2 \| K)$ et le compare ensuite à $Auth_i$. Si les valeurs ne sont pas égales, le serveur arrête la session. Sinon, l'objet intelligent est authentifié et le serveur S calcule $Auth_S = H(Id_{T_i} \| 1 \| K)$ et l'envoie à l'objet intelligent T_i.
4. À la réception, T_i calcule $Auth'_S = H(Id_{T_i} \| 1 \| K)$ et le compare à $Auth_S$. Si les valeurs ne sont pas égales, la session est arrêtée. Sinon, le serveur S est authentifié et donc l'authentification mutuelle est réalisée.

À la fin de la phase d'authentification, le serveur S et l'objet intelligent T_i partagent une clé de session, K, qui peut être utilisée pour garantir la confidentialité, l'intégrité et la vie privée pendant les échanges ultérieurs.

Analyse de sécurité

Dans ce qui suit, le protocole est analysé en ce qui concerne les exigences de sécurité définies en section 3.2 et selon d'autres propriétés supplémentaires (la disponibilité, la scalabilité et l'anonymat de l'objet intelligent).

- Authentification mutuelle :
  - — Authentification de l'objet intelligent : L'adversaire ne peut pas générer un message légitime $(R_i, Auth_i, M)$ d'un objet intelligent T_i, car il ne peut obtenir ni l'identifiant Id_{T_i} de l'objet intelligent T_i ni la clé de hachage associée HId_{T_i} ni la valeur aléatoire r_i, valeurs qui sont utilisées pour le calcul de $Auth_i$ et de M. Cependant, le serveur S peut de son côté calculer K (puisque $r_S \cdot R_i$ est égal à $r_i \cdot R_S$) qui lui permet d'obtenir HId'_{T_i} et vérifier si l'entrée existe réellement. Si l'entrée n'existe pas, le serveur S doit temporiser

avant d'arrêter définitivement la session. Autrement dit, il doit répondre dans un délai déterminé afin d'éviter de divulguer des informations à l'attaquant. Dans le cas où, l'entrée existe, S obtient Id'_{T_i} et le compare à Id_{T_i}. Toutefois, si la comparaison échoue, le serveur arrête la session. Dans le cas contraire, l'objet intelligent T_i est authentifié.

— Authentification du serveur : L'adversaire ne peut pas générer un message légitime $Auth_S$ de S, puisqu'il n'est pas en mesure d'obtenir l'identifiant de l'objet intelligent Id_{T_i} ou la valeur de K. Pour authentifier le serveur S, l'objet intelligent T_i doit calculer $Auth'_S$ pour vérifier ensuite si le $Auth_S$ reçu est valide ou pas.

- Confidentialité, intégrité et respect de la vie privée : À la fin du protocole, les deux entités S et T_i partagent une nouvelle clé de session K à laquelle les deux parties ont contribué puisqu'elle repose sur les valeurs aléatoires r_S généré par S et r_i généré par T_i pour chaque session. S et T_i peuvent l'utiliser pour dériver des clés permettant de garantir la confidentialité, l'intégrité et le respect de la vie privée pendant les échanges ultérieurs.
- Disponibilité :
  — Résistance aux attaques de désynchronisation : Le protocole proposé ne contenant pas d'étape de mise à jour de clé et celui-ci ayant été construit pour être « stateless », les clés ne peuvent pas être désynchronisées et le protocole peut être repris à partir de zéro à n'importe quelle étape de l'authentification.
  — Résistance aux attaques DoS : Le protocole proposé étant sans état, chaque transaction est considérée comme étant nouvelle. Du côté des objets intelligents, une nouvelle transaction d'authentification ne consomme pas plus d'espace mémoire que la précédente. Certes, le lancement d'une authentification utilise inévitablement des ressources de calcul. Le cas le plus défavorable du côté des objets intelligents dépend de l'implémentation et du type de la mémoire utilisée : si l'objet intelligent utilise une mémoire non volatile telle que la Flash ou l'EEPROM pour stocker ses calculs intermédiaires, les cellules mémoire peuvent mourir après 10^5 cycles d'écriture. En revanche, s'il utilise de la mémoire FRAM, elle durera jusqu'à 10^{12} cycles et s'il utilise de la RAM, le nombre de cycles sera infini. Du côté du serveur, puisque celui-ci doit être capable de gérer simultanément l'authentification de plusieurs objets intelligents, chaque nouvelle transaction d'authentification consomme certaines ressources en mémoire. Cependant, dans le cas d'une attaque par déni de service, notre mécanisme de recherche efficace limite beaucoup l'impact de l'attaque, puisque le serveur ne passe pas que peu de temps à faire des calculs et à interroger la base de données pour un faux HId'_{T_i}.
- Scalabilité : Grâce à l'efficacité de notre mécanisme de recherche proposé, le serveur S calcule d'abord K, puis vérifie si HId'_{T_i} existe dans sa base de données. Éventuellement, il véri-

fie la correspondance entre Id'_{T_i} et Id_{T_i} dans un délai bien déterminé, c'est-à-dire en $O(1)$. De plus, ce protocole permet l'authentification mutuelle entre des objets intelligents et plusieurs serveurs, à condition que ces derniers connaissent l'identité des objets intelligents. Cela signifie que si un serveur tombe en panne, l'authentification peut être effectuée par un autre serveur s'il connaît l'identité de l'objet intelligent et la clé de hachage de l'entrée associée.

- Anonymat de l'objet intelligent : Dans le protocole proposé, puisque l'identifiant de l'objet intelligent Id_{T_i} et la clé associée HId_{T_i} sont masqués en utilisant K et une fonction de hachage, un adversaire actif peut calculer K (en utilisant la valeur générée r_S et la valeur reçue R_i). De ce fait, il peut calculer HId'_{T_i}. Cependant, il ne peut pas calculer Id_{T_i}. Par conséquent, l'adversaire peut tracer les positions de l'objet intelligent, mais il ne peut pas obtenir son identité, ce qui garantit son anonymat.

⋆ Analyse formelle avec AVISPA ⋆

Comme le détail de la spécification est donné en annexe, nous ne présentons dans cette section que les points importants de la spécification du protocole et les résultats.

▷ Spécification du protocole

Le script HLPSL contenant les détails de notre protocole est présenté dans l'Annexe A.3. Il existe deux rôles de base S et T représentant respectivement le $Serveur$ et l'objet intelligent T_i. Le concept fondamental du protocole consiste à conserver les valeurs Id_{T_i}, HId_{T_i}, r_S et r_i `secret` pendant la phase d'authentification du protocole entre S et T. L'authentification mutuelle est réalisée via les objectifs `witness` et `request`.

▷ Résultats de la vérification

Les résultats après l'exécution de notre protocole codé en HLPSL sont donnés dans le Tableau 3.3.

TABLEAU 3.3 – Résultats de validation AVISPA

Moteur AVISPA	Résultat
$OFMC$	SAFE
$CL-AtSe$	SAFE
$SATMC$	INCONCLUSIVE
$TA4SP$	INCONCLUSIVE

Dans le Tableau 3.3, AVISPA retourne SAFE à partir de ses deux back-ends On-the-Fly Model-Checker (OFMC) et CL-based Model-Checker (CL-AtSe). Par contre SAT-based Model-Checker (SATMC) et Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) donnent INCONCLUSIVE en raison d'opérations non prises en charge, ce qui signifie que AVISPA ne peut pas trouver d'attaques contre notre protocole.

⋆ Analyse formelle avec ProVerif ⋆

Dans les paragraphes suivants, les points importants de la spécification du protocole et les résultats sont mis en avant.

▷ Spécification du protocole

Les scripts Pi Calculus appliqués sont donnés dans l'Annexe B.4. Comme pour AVISPA, il existe deux rôles de base `SERVERS` et `SOI` qui expliquent l'activité de $Server$ et T_i. Les valeurs secrètes sont vérifiées à l'aide de requêtes (« queries »). L'authentification mutuelle entre T_i et $Server$ est modélisée à l'aide d'événements mappés dans les sous-processus et les requêtes de `SOI` et de `SERVERS`. Les détails sont fournis en annexe.

▷ Résultats de la vérification

Les résultats sont disponibles dans l'Annexe B.5. Ils montrent que la confidentialité (« secrecy ») de `idt`, `hidt`, `ri`, `rs` est préservée par le protocole et que l'authentification mutuelle entre `SERVERS` et `SOI` est réalisée.

3.4.3 La description du protocole d'authentification M2M basé sur ECC

Dans cette section, nous présentons et validons notre nouveau protocole d'authentification M2M basé sur ECC. De la même façon que pour les protocoles précédents, il comprend deux phases : une phase de configuration et une phase d'authentification.

Phase de configuration

Dans cette phase, le serveur de l'autorité de certification génère et partage avec les autres entités les paramètres du système (q, a, b, p, P) à utiliser pour la courbe elliptique E. Ensuite, chaque objet intelligent T_i calcule ses clés privée/publique (x_i, P_i). De même, le serveur de l'autorité de certification S_{AC} calcule ses clés privée/publique (x_S, P_S).

Ensuite, chaque objet intelligent T_i demande au serveur S_{AC} de certifier sa clé publique à l'aide d'une signature de Schnorr [145] : S_{AC} choisit une valeur aléatoire $s_i \in \mathbb{Z}_p^*$ et calcule $\bar{P}_i = s_i \cdot P$ et $z_i = s_i + H(P_i, \bar{P}_i) \cdot x_S$. La signature résultante $\sigma_i = (\bar{P}_i, z_i)$ est fournie au dispositif intelligent T_i.

À la fin de cette phase, l'objet intelligent T_i stocke dans sa mémoire les paramètres du système, sa paire de clés (x_i, P_i), la signature σ_i de sa clé publique et la clé publique du serveur P_S.

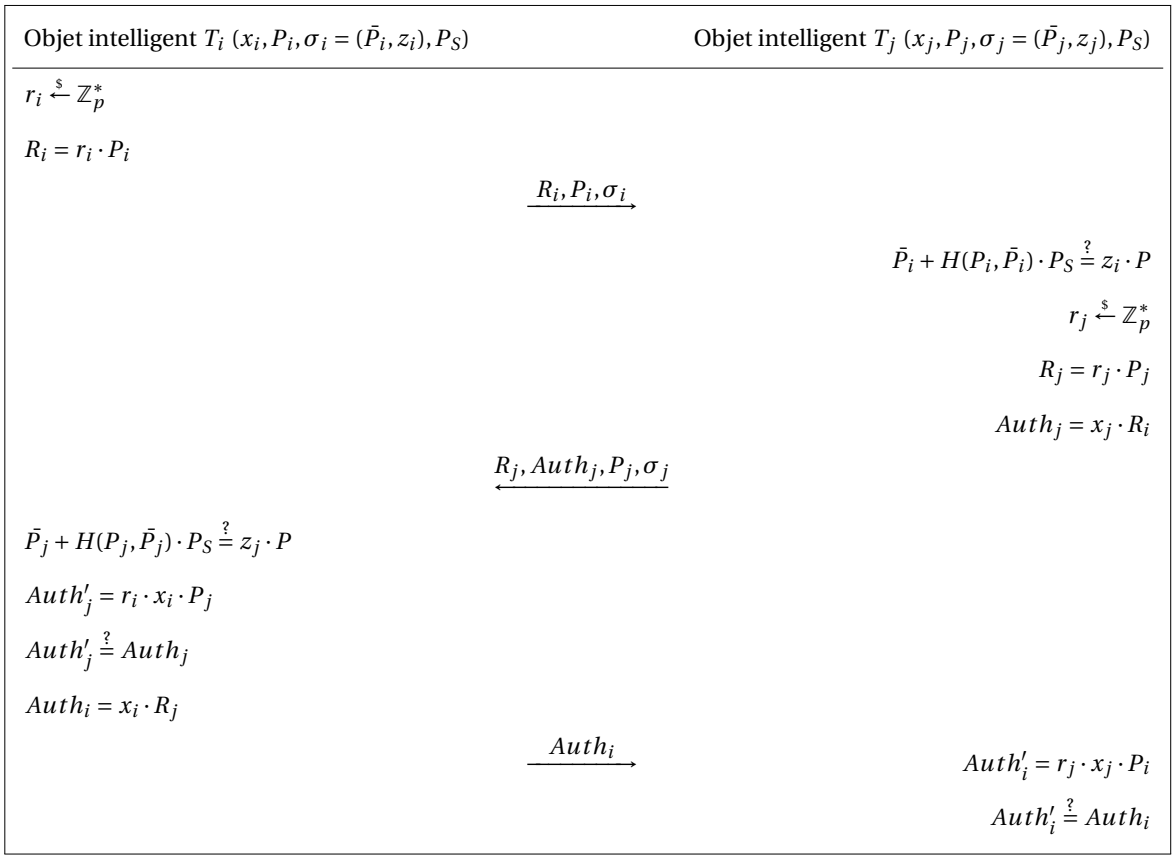


FIGURE 3.4 – Protocole d'authentification M2M basé sur ECC

Phase d'authentification

Le protocole d'authentification M2M proposé permet l'authentification entre n'importe quels objets intelligents qui souhaitent communiquer entre eux de manière sécurisée. Sur la Figure 3.4, nous présentons les étapes permettant l'authentification mutuelle entre deux objets intelligents.

1. Le premier objet intelligent T_i qui entame la communication choisit d'abord une valeur aléatoire $r_i \in \mathbb{Z}_p^*$ et calcule $R_i = r_i \cdot P_i$. Ensuite, T_i envoie (R_i, P_i, σ_i) au deuxième objet intelligent T_j.
2. À la réception, T_j vérifie d'abord la signature de Schnorr afin de valider la clé publique P_i reçue. Si la validation échoue, T_j arrête la session. Sinon, T_j choisit une valeur aléatoire $r_j \in \mathbb{Z}_p^*$ et calcule $R_j = r_j \cdot P_j$ et $Auth_j = x_j \cdot R_i$. Ensuite, T_j envoie le message $(R_j, Auth_j, P_j, \sigma_j)$ à T_i.
3. T_i vérifie alors la signature de Schnorr reçue de la clé publique de T_j. Si la vérification échoue, T_i arrête la session. Sinon, T_i calcule $Auth'_j = r_i \cdot x_i \cdot P_j$ et vérifie si la valeur de

$Auth_j$ reçue correspond. Si elles sont différentes, T_i arrête la session. Sinon, T_j est authentifié auprès de T_i. Enfin, T_i calcule $Auth_i = x_i \cdot R_j$ et l'envoie à T_j.

4. À la réception, T_j calcule $Auth'_i = r_j \cdot x_j \cdot P_i$ et vérifie si le $Auth_i$ reçu correspond. S'ils ne sont pas égaux, T_j arrête la session. Sinon, T_i est authentifié auprès de T_j et l'authentification mutuelle est assurée.

Pour générer une clé de session partagée K, chacun des deux dispositifs intelligents T_i et T_j doit faire respectivement les calculs $r_i \cdot R_j$ et $r_j \cdot R_i$. Ce processus coûte à chaque objet intelligent une multiplication scalaire supplémentaire.

Analyse de sécurité

Dans ce qui suit, le protocole est analysé en ce qui concerne les exigences de sécurité définies en section 3.2 et selon certaines propriétés supplémentaires (la disponibilité, la scalabilité et l'anonymat des objets intelligents).

- Authentification mutuelle :
  - Authentification de T_j : L'adversaire ne peut pas générer un message légitime ($R_j, Auth_j, P_j, \sigma_j$) d'un objet intelligent T_j, puisqu'il ne peut pas obtenir la clé privée x_j (appartenant à T_j) utilisée pour le calcul de $Auth_j$. Bien qu'il puisse obtenir les valeurs légitimes R_j, P_j et σ_j, lorsque T_i calcule $Auth'_j$ et le compare à la valeur reçue $Auth_j$, ils ne peuvent pas être similaires si la clé privée x_j n'a pas été utilisée pour le calcul de $Auth'_j$, puisque $Auth'_j = r_i \cdot x_i \cdot P_j = r_i \cdot x_i \cdot (x_j \cdot P) = r_i \cdot x_j \cdot (x_i \cdot P) = r_i \cdot x_j \cdot P_i = x_j \cdot (r_i \cdot P_i) = x_j \cdot R_i = Auth_j$. Ainsi, si $Auth_j$ et $Auth'_j$ sont différents, l'objet intelligent T_j n'est pas authentifié.
  - Authentification de T_i : Des explications similaires s'appliquent pour l'authentification de T_i.
- Confidentialité, intégrité et respect de la vie privée : À la fin du protocole, les deux entités T_i et T_j peuvent générer une nouvelle clé de session partagée K en calculant respectivement $r_i \cdot R_j$ et $r_j \cdot R_i$ à laquelle les deux ont contribué, car elle est basée sur les valeurs aléatoires r_i et r_j pour chaque session. T_i et T_j peuvent l'utiliser pour dériver des clés afin de permettre de garantir la confidentialité, l'intégrité et le respect de la vie privée des échanges ultérieurs.
- Disponibilité :
  - Résistance aux attaques de désynchronisation : Le protocole proposé ne contenant pas d'étape de mise à jour de clé et celui-ci ayant été construit pour être « stateless », les clés ne peuvent pas être désynchronisées et le protocole peut être repris depuis le début à n'importe quelle étape de l'authentification.

— Résistance aux attaques DoS : Le protocole proposé étant sans état, chaque transaction est considérée comme étant nouvelle. Pour chaque entité, une nouvelle transaction d'authentification ne consomme pas plus d'espace mémoire que la précédente. Elle utilise certes des ressources de calcul mais c'est inévitable. Le cas le plus défavorable dépend de l'implémentation et du type de la mémoire utilisée : si les objets intelligents utilisent une technologie de mémoire non volatile comme la Flash ou l'EEPROM pour stocker leurs calculs intermédiaires, les cellules de la mémoire peuvent mourir après 10^5 cycles d'écriture. Cependant, s'ils utilisent de la mémoire FRAM, elle durera jusqu'à 10^{12} cycles et s'ils utilisent de la RAM, le nombre de cycles sera infini.

- Scalabilité : Tant que S_{AC} est disponible pour la phase de configuration, il est possible d'ajouter de nouveaux objets intelligents au système. Lorsqu'ils sont dans le système, étant donné que les objets intelligents s'authentifient de manière décentralisée, le système est scalable.
- Anonymat des objets intelligents : Dans le protocole proposé, étant donné que P_i et σ_i sont toujours identiques, il est possible de tracer un objet intelligent et de l'identifier (si nous considérons une clé publique comme un identifiant), mais ce n'est pas l'une de nos exigences. Malgré ce problème, la confidentialité et le respect de la vie privée des échanges ultérieurs entre objets intelligents sont elles assurées grâce à la clé partagée K.

⋆ Analyse formelle avec AVISPA ⋆

Comme le détail de la spécification est donné en annexe, nous ne présentons dans cette section que les points importants de la spécification du protocole et les résultats.

▷ Spécification du protocole

Le script HLPSL de notre protocole est présenté dans l'Annexe A.4. Il existe deux rôles de base A et B qui correspondent à l'activité des nœuds T_i et T_j. Le concept fondamental du protocole consiste à conserver les valeurs x_i, x_j, r_i et r_j `secret` tout au long de la phase d'authentification du protocole entre T_i et T_j. L'authentification mutuelle est réalisée via les objectifs `witness` et `request`. Les détails sont fournis en annexe.

▷ Résultats de la vérification

Les résultats après l'exécution de notre protocole codé en HLPSL sont donnés dans le Tableau 3.4.

Dans le Tableau 3.4, AVISPA retourne SAFE à partir de ses deux back-ends On-the-Fly Model-Checker (OFMC) et CL-based Model-Checker (CL-AtSe). Par contre SAT-based Model-Checker (SATMC) et Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) donnent INCONCLUSIVE en raison d'opérations non prises en charge, ce qui signifie

TABLEAU 3.4 – Résultats de validation AVISPA

Moteur AVISPA	Résultat
$OFMC$	SAFE
$CL-AtSe$	SAFE
$SATMC$	INCONCLUSIVE
$TA4SP$	INCONCLUSIVE

que AVISPA ne peut pas trouver d'attaques contre notre protocole.

⋆ Analyse formelle avec ProVerif ⋆

Dans les paragraphes suivants, les points importants de la spécification du protocole et les résultats sont mis en avant.

▷ **Spécification du protocole**

Les scripts Pi Calculus appliqués sont donnés dans l'Annexe B.6. Comme pour AVISPA, il existe deux rôles de base `SOA` et `SOB` qui expliquent l'activité de T_i et T_j. Les valeurs secrètes sont vérifiées à l'aide de requêtes (« queries »). L'authentification mutuelle entre T_i et T_j est modélisée à l'aide d'événements mappés dans les sous-processus et les requêtes de `SOA` et de `SOB`. Les détails sont fournis en annexe.

▷ **Résultats de la vérification**

Les résultats sont disponibles dans l'Annexe B.7. Ils montrent que la confidentialité (« secrecy ») de `xa`, `xb`, `ra` et `rb` est préservée par le protocole et l'authentification mutuelle entre `SOA` est `SOB` est réalisée.

3.5 L'évaluation des protocoles

Dans le but d'évaluer l'efficacité de nos propositions, nous avons opté pour une analyse formelle en fonction des paramètres du domaine des courbes utilisées et du nombre d'objets intelligents dans le système au lieu de choisir arbitrairement des objets intelligents particuliers. Cependant pour évaluer les performances de nos propositions, nous avons mesuré le temps réel pris par l'opération la plus coûteuse (la multiplication scalaire) présent dans nos différents protocoles pour différentes courbes sur le dispositif Multos IoT Trust Anchor. Ce dernier est limité en ressources et exécute un système d'exploitation sécurisé sur un microcontroleur lui-même sécurisé.

Afin de confirmer nos résultats, nous avons également effectué des mesures sur un second dispositif utilisant un modèle de microcontroleur similaire mais qui exécute la technologie Java Card sur un facteur de forme de type carte à puce à contact.

3.5.1 L'analyse des performances

Dans cette section, nous analysons les performances des protocoles proposés en fonction des coûts de calcul, des coûts de communication et des besoins en stockage. Dans ce cadre, nous considérons un système composé de n objets intelligents. Selon les paramètres du domaine de la courbe elliptique E choisie, la taille des points sur E et la taille des scalaires utilisés varient en fonction de la formule suivante : la taille d'un point en bits est $2N$ et la taille d'un scalaire est N. Généralement, les paramètres du domaine (q, a, b, p, P) ont la taille de $6N$ bits (les tailles en bits de q, a et b sont souvent similaires à celles de p c'est-à-dire N et le générateur P est un point de E donc sa taille en bits est $2N$). Nous supposons que Id_{T_i} et $H(Id_{T_i})$ sont des scalaires, donc la taille en bits de chacun est N bits.

T_{sm} et T_{add} désignent les temps que prennent respectivement une multiplication scalaire et une addition de points.

3.5.1.1 Analyse de performances des protocoles d'authentification M2C

Dans cette section, nous comparons nos propositions M2C à une sélection de protocoles (présentés dans la section 2.6) tels que Dinarvand et al.'s [158], Liao et al.'s [157], Zhao's [155], Alamr et al.'s [156] et Jin et al.'s [154].

3.5.1.1.1 Coût de calculs

Les coûts de calculs pour les objets intelligents et pour le serveur lors de la phase d'authentification des protocoles proposés et celles des autres protocoles sélectionnés sont respectivement résumés dans les Tableaux 3.5 et 3.6 .

Le Tableau 3.5 montre que les meilleurs protocoles en termes de coût de calculs pour les objets intelligents sont nos deux propositions : 1) le protocole d'authentification basé sur les fonctions de hachage et 2) notre amélioration du protocole de Jin et al. Par rapport à Jin et al. nous gagnons du côté des objets intelligents, mais perdons du côté serveur en ce qui concerne la multiplication scalaire.

Le Tableau 3.6 montre que les meilleurs protocoles en termes de coût de calcul pour le serveur sont aussi nos deux propositions : 1) le protocole d'authentification basé sur les fonctions de hachage et 2) notre amélioration du protocole de Jin et al. Même si nous avons une multiplication scalaire en plus, nous avons réduit la complexité de la recherche dans la base de données à O(1).

TABLEAU 3.5 – Comparaison des coûts de calculs des objets intelligents pour différents protocoles M2C

Protocole	**Coût de calculs**
Liao et al. [157]	$5 * T_{sm} + 3 * T_{add}$
Zhao [155]	$5 * T_{sm} + 3 * T_{add}$
Alamr et al. [156]	$4 * T_{sm} + 1 * T_{add}$
Dinarvand et al. [158]	$3 * T_{sm} + 2 * T_{add}$
Jin et al. [154]	$4 * T_{sm} + 1 * T_{add}$
Notre amélioration de Jin et al.	$3 * T_{sm} + 1 * T_{add}$
Notre proposition d'authentification basée sur les fonctions de hachage	$2 * T_{sm}$

TABLEAU 3.6 – Comparaison des coûts de calculs du serveur pour différents protocoles M2C

Protocole	**Coût de calculs**	**Complexité de la recherche dans la base de données**
Liao et al. [157]	$5 * T_{sm} + 3 * T_{add}$	$O(n)$
Zhao et al. [155]	$5 * T_{sm} + 3 * T_{add}$	$O(n)$
Alamr et al. [156]	$5 * T_{sm} + 1 * T_{add}$	$O(n)$
Dinarvand et al. [158]	$3 * T_{sm} + 2 * T_{add}$	$O(n)$
Jin et al. [154]	$2 * T_{sm}$	$O(n)$
Notre amélioration de Jin et al.	$3 * T_{sm}$	$O(1)$
Notre proposition d'authentification basée sur les fonctions de hachage	$2 * T_{sm}$	$O(1)$

Dans nos deux propositions M2C (c'est-à-dire amélioration du protocole de Jin et al. et du protocole d'authentification basé sur les fonctions de hachage), nous avons amélioré le temps de transaction de la base de données côté serveur pendant l'authentification d'un objet intelligent en réduisant la complexité de recherche dans la base de données à $O(1)$ alors qu'elle est de l'ordre de $O(n)$ pour tous les autres protocoles.

3.5.1.1.2 Coûts de communication

Pour déterminer les coûts de communication, nous avons calculé la taille totale en bits des messages transmis au cours de la phase d'authentification et nous avons également compté le nombre d'échanges.

Dans le Tableau 3.7, les deux meilleurs protocoles en termes de coût de communication du serveur sont : 1) le protocole d'authentification basé sur les fonctions de hachage ; et 2) la version originale du protocole de Jin et al. étant donné que dans notre amélioration de ce dernier, nous avons décidé de calculer la multiplication scalaire au niveau du serveur à la place de l'objet intelligent (c'est-à-dire qu'un point est envoyé au lieu d'un scalaire). Cependant, il serait possible de

TABLEAU 3.7 – Comparaison des coûts de communication pour différents protocoles M2C

Protocole	**SO→S (bits)**	**S→SO (bits)**	**Total (bits)**	**# d'échanges**
Liao et al. [157]	$4N$	$4N$	$8N$	3
Zhao [155]	$4N$	$4N$	$8N$	3
Alamr et al. [156]	$4N$	$6N$	$10N$	3
Dinarvand et al. [158]	$5N$	$4N$	$9N$	4
Jin et al. [154]	$4N$	$3N$	$7N$	3
Notre amélioration de Jin et al.	$4N$	$4N$	$8N$	3
Notre proposition d'authentification basée sur les fonctions de hachage	$4N$	$3N$	$7N$	3

transmettre seulement la coordonnée x de $Auth_S$ sans compromettre la sécurité ce qui le rendrait équivalent.

3.5.1.1.3 Besoins en stockage

Le besoin en stockage est défini comme l'espace utilisé pour stocker les données dans les entités communicantes du système. Dans n'importe quel protocole, cela représente les paramètres du domaine et les données supplémentaires (telles que les informations d'identification, les clés partagées et les identités) stockées à la fin de la phase de configuration.

TABLEAU 3.8 – Comparaison des besoins de stockage pour différents protocoles M2C

Protocole	**SO (bits)**	**S (bits)**
Liao et al. [157]	$6N+5N$	$6N+3N+3N\cdot n$
Zhao [155]	$6N+5N$	$6N+3N+3N\cdot n$
Alamr et al. [156]	$6N+5N$	$6N+3N+2N\cdot n$
Dinarvand et al. [158]	$6N+5N$	$6N+3N+3N\cdot n$
Jin et al. [154]	$6N+4N$	$6N+3N+2N\cdot n$
Notre amélioration de Jin et al.	$6N+4N$	$6N+3N+2N\cdot n$
Notre proposition d'authentification basée sur les fonctions de hachage	$6N+2N$	$6N+2N\cdot n$

Dans le Tableau 3.8, les trois meilleurs protocoles en termes de stockage pour les objets intelligents et pour les serveurs sont : 1) notre protocole d'authentification basé sur les fonctions de hachage ; ensuite, à égalité, 2) notre amélioration du protocole de Jin et al. et l'original du protocole de Jin et al. Nous n'avons pas pris en compte le protocole d'Alamr et al. en raison des exigences de stockage sur l'objet intelligent.

3.5.1.1.4 Résumé

Selon les critères de performances analysés, les deux protocoles les plus efficaces sont les deux protocoles M2C que nous avons proposés. Le protocole d'authentification basé sur les fonctions de hachage surpasse tous les autres dans tous les critères. Bien que nos améliorations du protocole de Jin et al. signifient qu'il y a un calcul supplémentaire sur le serveur, cet inconvénient est négligeable comparé au temps gagné en introduisant notre méthode pour réduire l'opération de recherche dans la base de données qui passe d'une complexité linéaire ($O(n)$) à une complexité constante ($O(1)$). Nous avons également expliqué comment le coût de communication d'un serveur vers un objet intelligent peut être réduit à celui du protocole original de Jin et al. sans compromettre sa sécurité.

Pour conclure, le protocole d'authentification basé sur les fonctions de hachage est la meilleure solution pour un mode de communication M2C, car il est plus flexible et plus évolutif que les autres protocoles que nous avons examinés.

3.5.1.2 Analyse du protocole d'authentification M2M

Comme expliqué précédemment dans la section 2.6, il n'existe aucun protocole d'authentification similaire (connexion directe, c'est-à-dire sans proxy) proposé dans la littérature et nous ne pouvons donc qu'analyser notre proposition.

Contrairement au mode de communication M2C, dans lequel l'objet intelligent communique uniquement avec le serveur, dans les protocoles d'authentification M2M, un objet intelligent doit généralement disposer d'informations de base sur les autres objets intelligents présents dans le système afin de communiquer avec eux. De ce fait, dans notre protocole d'authentification basé sur ECC pour réduire les besoins en stockage des objets intelligents et permettre la communication dynamique avec de nouveaux objets intelligents, en plus de leurs propres informations d'identification, sur chaque objet intelligent nous ne stockons que la clé publique du serveur de l'autorité de certification. Cela permet à l'objet intelligent de vérifier l'authenticité des clés publiques des autres objets intelligents (puisqu'elles ont été signées par S_{AC}) sans avoir besoin de les stocker dans sa mémoire.

L'analyse des performances fournie dans le Tableau 3.9 montre que le protocole proposé est parfaitement adapté à un mode de communication M2M.

3.5.2 L'implémentation et discussion des résultats

Pour évaluer la performance pratique de nos propositions, nous avons effectué des mesures de temps des différentes opérations sur deux cibles : le Multos IoT Trust Anchor [172] et Universal

TABLEAU 3.9 – Analyse des performances du protocole d'authentification basé sur ECC par objet intelligent pour notre proposition de protocole M2M

Coût de calculs	$5 * T_{sm} + 1 * T_{add}$
Coût de communication (bits)	$9N$
Besoin en stockage (bits)	$6N + 8N$

JCard [173]. Multos IoT Trust Anchor étant conçu pour les applications IoT, la plupart des opérations utilisées dans les protocoles proposés ont été soumises à des mesures de performance : hachage (SHA-1), xor, multiplication, multiplication modulaire et multiplication scalaire. Par contre, nous n'avons utilisé la seconde cible, Universal JCard, que pour confirmer que l'opération la plus coûteuse, à savoir la multiplication scalaire, pouvait offrir les mêmes performances sur la même plateforme matérielle mais sur un autre système d'exploitation.

3.5.2.1 Choix des outils

Nous avons décidé de réaliser nos mesures expérimentales sur les deux plates-formes susmentionnées pour plusieurs raisons. Les deux dispositifs sont des plates-formes avec des ressources très contraintes composées d'un circuit intégré (ICC) sécurisé et d'un système d'exploitation également sécurisé. Multos IoT Trust Anchor est la première initiative du consortium Multos [174] pour porter le système d'exploitation pour carte à puce, Multos, sur un système circuit intégré sécurisé pour le domaine de l'IoT afin de permettre l'exécution d'applications sécurisées. Le matériel utilisé pour cette plate-forme étant un modèle SLE78CLUFX5000PHM [175] de la série SLE78 d'Infineon qui est souvent utilisé dans les cartes à puce, nous avons ainsi choisi d'étudier les performances de l'opération cryptographique la plus coûteuse, c'est-à-dire la multiplication scalaire, sur un autre modèle, le SLE78CLFX4000P [176] de la série des SLE78, qui exécute une implémentation du système d'exploitation Java Card. Notre décision a également été motivée par le choix récent d'Oracle Inc. de cibler les dispositifs IoT avec la technologie Java Card [177], comme le montre la version 3.1 des spécifications de la Java Card [178, 179]. Les deux circuits intégrés SLE78 contiennent plusieurs capteurs matériels de sécurité (température, lumière, tension, fréquence) et protections (I^2-shield, technique de blindage spécifique à Infineon associée à un câblage sécurisé des signaux de sécurité critiques afin d'éviter toute manipulation ou toute écoute passive par un attaquant). Ces contrôleurs de sécurité de 16-bits intègrent l'Integrity Guard [180] qui, en résumé consiste à ce que chacun des deux CPU vérifient mutuellement en continue si l'autre unité fonctionne correctement, et le SOLID FLASH™ [181] qui est une technologie qui remplace le masquage de la mémoire ROM par une mémoire Flash afin de fournir une approche flexible pour ré-

pondre aux besoins du marché en réduisant les temps de développement.

3.5.2.1.1 *Multos IoT Trust Anchor*

Multos IoT Trust Anchor consiste en un ICC sécurisé, un SLE78, qui exécute le système d'exploitation Multos. Ce système d'exploitation est connu pour être l'un des plus sûrs du secteur des cartes à puce : en d'autres termes, les opérations sensibles sont effectuées selon les techniques de l'état de l'art (par exemple, les opérations sont effectuées en « temps constant » pour éviter les fuites d'informations). La cible est disponible pour les développeurs sous deux formats [182] : un format DIP32 pouvant être branché sur une carte de développement pour donner accès aux interfaces GPIO, I2C, SPI, interfaces de carte à puce série, contact et sans contact ; ainsi qu'une carte au format nano qui fournit dans un format plus compact toutes les broches pour ces mêmes interfaces. Les deux formats exécutent respectivement le M5-P22 (aka Multos 4.5.3) [183] et le M5-P19 (aka Multos 4.5.1) [184], des versions d'implémentations de l'OS Multos. La carte de développement pour le format DIP32 offre différentes fonctionnalités qui facilitent le développement : un régulateur intégré 3,3 V (pour alimentation par USB ou par une alimentation DC externe), un interrupteur on/off, une interface USB vers port série, les LEDs pour Tx/Rx, une LED tout usage, un bouton-poussoir et des broches pour accéder aux différentes interfaces. Dans ce qui suit, nous avons utilisé le format DIP32 branché à la carte de développement. Pour développer des applications, nous avons utilisé le Multos SmartDeck [185], ce qui nous a permis de compiler facilement nos programmes en langage C pour les cibles Multos.

3.5.2.1.2 *Universal JCard*

L'Universal JCard est disponible au format carte à puce avec des interfaces avec contact ISO7816 et sans contact ISO14443. Il s'agit d'un ICC sécurisé, SLE78, qui exécute le système d'exploitation Java Card. Java Card [186] est une technologie multi-applicative pour les dispositifs à mémoire limitée qui permet d'exécuter en toute sécurité des applications de différents fournisseurs [187]. L'implémentation disponible sur Universal JCard est l'implémentation de référence d'Oracle de Java Card en version 3.0.1. Pour développer des applications, appelées applets, nous avons utilisé le logiciel de développement d'Infineon [188], JCIDE qui signifie "Java Card IDE" et qui s'appuye sur le kit de développement de Java Card [189] pour construire le fichier de distribution, c'est-à-dire le fichier `cap`. Nous l'avons également utilisé pour charger ce fichier sur la carte en utilisant l'outil intégré compatible à GlobalPlatform (standard qui définit entre autres comment charger une application sur une carte multi-applicative).

Il est à noter que l'interaction avec une carte à puce se limite à l'envoi de commandes par un lecteur et à la réception de réponses provenant de la carte selon le protocole APDU (que ce soit

sur les interfaces à contact ou sans contact).

3.5.2.2 Méthodologie pour mesurer les performances sur Multos IoT Trust Anchor

Pour mesurer le temps interne pris par le système d'exploitation Multos de notre dispositif pour exécuter les opérations de fonction de hachage, xor, multiplication, multiplication modulaire et multiplication scalaire, nous avons utilisé des fonctions de l'API. Comme dans notre implémentation, il n'était pas possible d'accéder directement à la primitive de multiplication scalaire et que seules les courbes de Short-Weierstrass étaient prises en charge avec en plus des contraintes sur la spécification des paramètres de domaine, nous n'avons effectué des mesures de temps que pour plusieurs courbes elliptiques (secp192k1, secp192r1, Wei22519, secp256k1, secp256r1, secp384r1 et secp521r1).

En résumé, nous avons implémenté une application très simple appelant des fonctions de l'API, utilisant la fonction de hachage `multoSHA1`, xor `multosXor`, multiplication `multosMultiply` et multiplication scalaire `multosECDH`. Comme la multiplication modulaire n'était pas directement accessible via les fonctions de l'API, nous avons implémenté une version naïve utilisant `multosMultiply` et `multosDivide`. Cette version n'est pas optimisée et la taille de l'opérande est limité à un maximum de 512 bits. Nous avons exécuté notre application sur les paramètres du domaine supportés pour les courbes elliptiques (pour avoir des résultats pour la multiplication scalaire). Pour les autres opérations, nous les avons exécutées sur des opérandes dont la longueur était de N ou $2N$ puisque c'est la taille de données la plus utilisée dans les protocoles étudiés.

Pour obtenir le temps le plus précis pour chaque opération, nous avons utilisé une broche GPIO pour déclencher et arrêter nos mesures. Nous l'avons fixé à la valeur `HIGH` juste avant l'appel de la fonction, puis sur `LOW` juste après le retour de l'appel. Comme illustré Figure 3.5, nous avons connecté le canal 1 de notre oscilloscope à la broche GPIO (ici GPIO7) utilisée comme déclencheur et à la broche GND. Nous avons évidemment vérifié que le temps nécessaire pour faire changer d'état la broche GPIO de `LOW` à `HIGH`, puis à `LOW` était négligeable.

Avec cette configuration matérielle et après avoir configuré l'oscilloscope pour qu'il se déclenche sur un front montant du canal 1 pour démarrer la capture du signal sur ce canal, nous avons pu obtenir certaines mesures comme celle illustrée Figure 3.6 qui est une multiplication scalaire sur secp256r1.

3.5.2.3 Résultats de performance sur Multos IoT Trust Anchor

Les mesures de temps que nous reportons dans le Tableau 3.10 pour les différentes courbes correspondent aux intervalles de temps mesurés entre le front montant et le front descendant. Ce n'est pas une approximation exacte du temps de traitement de la multiplication scalaire, car l'ap-

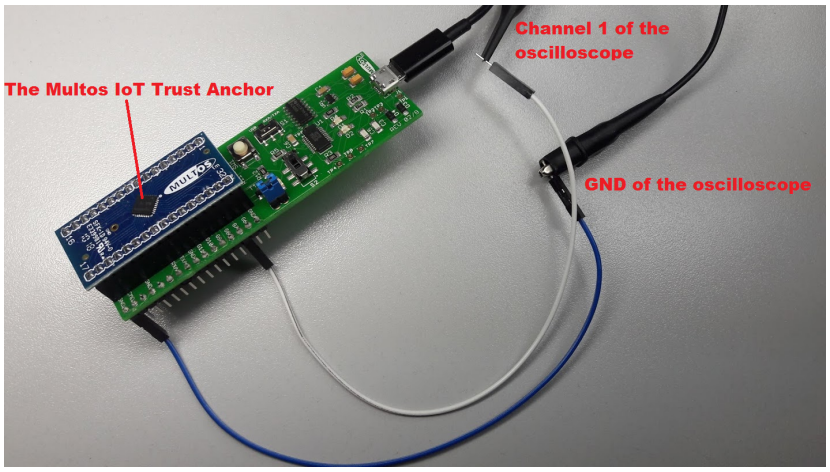


FIGURE 3.5 – Multos IoT Trust Anchor exécutant notre application et connecté à la voie 1 de notre oscilloscope

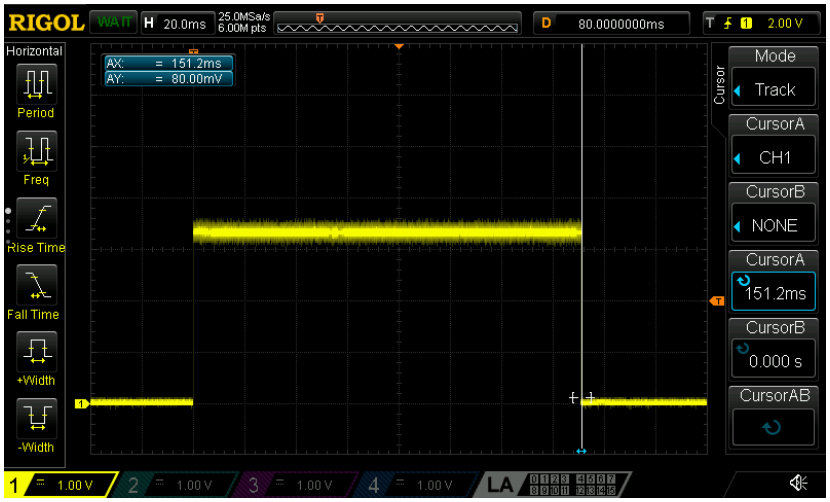


FIGURE 3.6 – Temps pour une multiplication scalaire sur secp256r1 s'exécutant sur Multos IoT Trust Anchor

pel d'API peut certainement contenir des opérations supplémentaires (par exemple, des vérifications), mais leur temps d'exécution est négligeable. Comme indiqué dans le Tableau 3.10, même pour les paramètres de domaine de même taille en bits, les temps de multiplication scalaire sont différents. Cependant, pour la même courbe, nous obtenons toujours les mêmes résultats lorsque nous multiplions des scalaires et des points différents dans la multiplication scalaire. Ceci est certainement dû aux techniques de programmation en temps constant utilisées par les développeurs du système d'exploitation sécurisé afin d'éviter les fuites d'informations. Pour rendre les mesures de multiplication scalaire comparables au reste de nos mesures présentées dans le Tableau 3.11, nous avons décidé d'utiliser un temps moyen pour la taille d'un paramètre de domaine (c'est-à-dire la taille de la clé privée).

Le Tableau 3.11 résume les résultats obtenus pour les différentes opérations et ils sont présen-

TABLEAU 3.10 – Temps interne consacré à la multiplication scalaire sur différentes courbes sous Multos IoT Trust Anchor (ms)

Taille de la clé privée (bits)	Nom de la courbe	Temps (ms)	Temps moyen (ms)
192	secp192k1	108,2	107,8
	secp192r1	107,4	
256	Wei25519	134,2	147,27
	secp256k1	156,4	
	secp256r1	151,2	
384	secp384r1	285	285
521	secp521r1	485	485

tés Figure 3.7.

TABLEAU 3.11 – Temps interne consacré à l'exécution des différentes opérations sur différentes tailles de données sur Multos IoT Trust Anchor (ms)

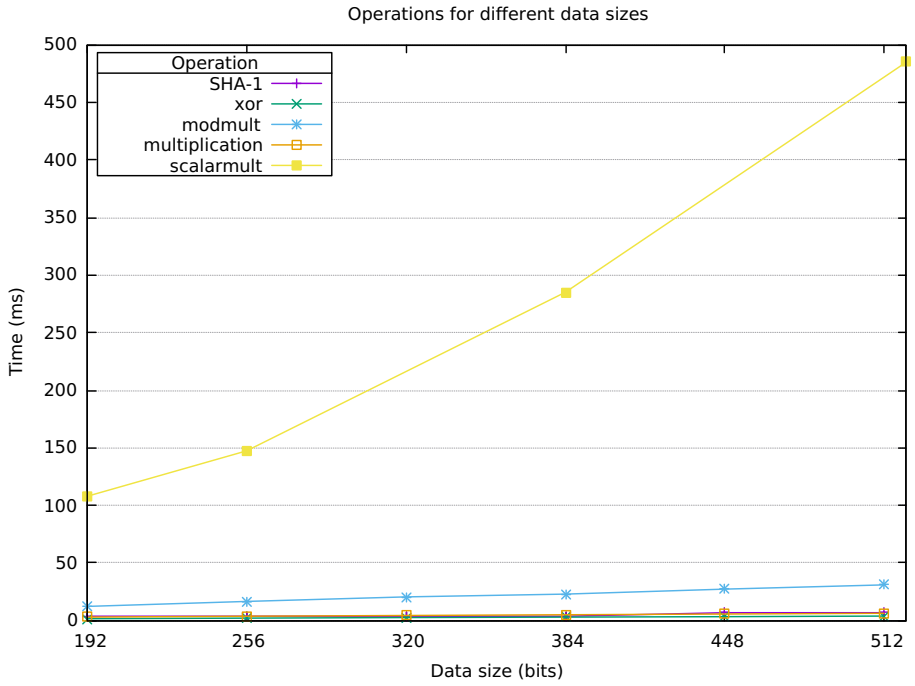
Taille des donnés (bits) / Opération	192	256	320	384	448	512	521
SHA-1	3,72	3,72	3,72	3,72	6,72	6,80	
xor	1,55	2,09	2,54	2,96	3,37	3,84	
multiplication	2,91	3,70	4,32	5,02	5,63	6,28	
modmult (software implementation)	12,24	16,56	20,44	22,90	27,20	30,90	
scalarmult (mean)	107,8	147,27		285			485

La Figure 3.7 confirme clairement que l'opération la plus consommatrice de temps à considérer dans nos protocoles est la multiplication scalaire. Les courbes de performance de chaque opération sont représentées par un axe des ordonnées, une échelle linéaire sur l'une et une échelle logarithmique sur la seconde.

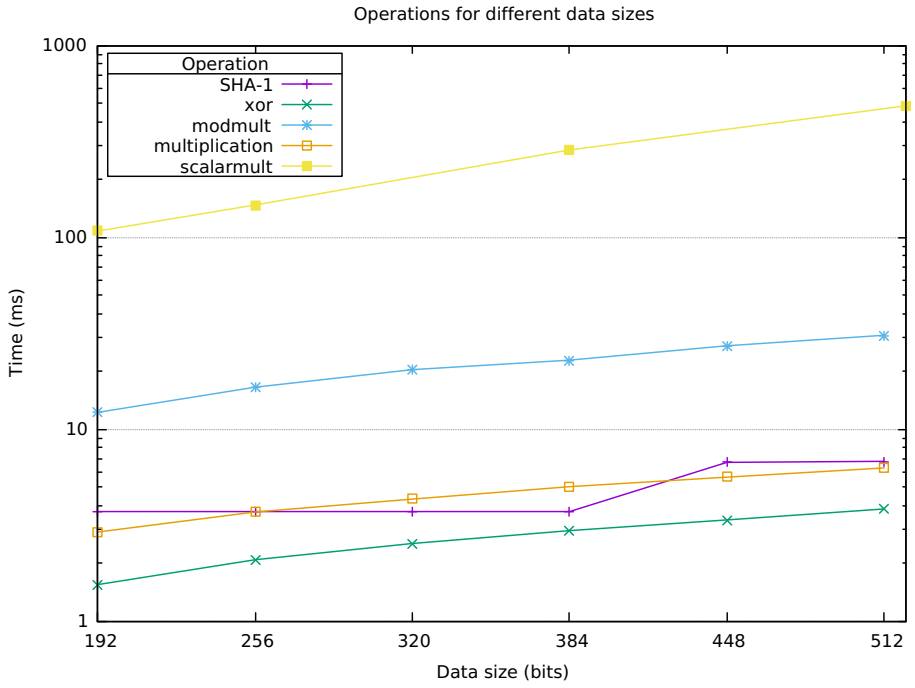
Il est à noter que nos résultats ont été les premiers à être publiés sur la plate-forme Multos IoT Trust Anchor dans notre article [190].

3.5.2.4 Méthodologie pour mesurer les performances sur Universal JCard

Pour mesurer le temps interne de la multiplication scalaire sur Universal JCard, nous avons utilisé les fonctions de l'API. Comme Java Card prend en charge les courbes de Short-Weierstrass



Performance where y-axis has a linear scale



Performance where y-axis has a log scale

FIGURE 3.7 – Performances des différentes opérations nécessaires pour les protocoles étudiés exécutées sur Multos IoT Trust Anchor

et que l'implémentation Universal JCard prend également en charge diverses tailles de clé jusqu'à 521 bits, nous avons effectué des mesures temporelles pour plusieurs courbes elliptiques (secp112r1, secp128r1, secp160k1, secp160r1, secp192k1, secp192r1, secp224k1, Wei22519, secp256-k1, secp256r1, brainpoolP256r1, secp384r1 et secp521r1).

En résumé, nous avons implémenté une applet très simple créant des conteneurs de clés et nous avons envoyé des commandes APDU pour configurer les différents paramètres du domaine de ces clés (et donc des courbes). Pour mesurer une multiplication scalaire unique, nous avons créé un objet `KeyAgreement` du type `KeyAgreement.ALG_EC_SVDP_DH_PLAIN`, c'est-à-dire, la primitive Diffie-Hellmann de dérivation de valeur secrète. Nous avons initialisé cet objet avec une clé privée aléatoire et demandé la dérivation de la valeur secrète en appelant la méthode `generateSecret` sur le point P qui est le générateur de la courbe choisie : cet appel est simplement une multiplication scalaire de la clé privée (un scalaire) et de P, un point. Pour éviter les imprécisions dues à des facteurs tels que l'overhead et les mutex sur l'ordinateur liés au système d'exploitation et aussi pour minimiser les retards de communication entre l'ordinateur et la carte à puce, nous avons décidé d'utiliser un oscilloscope pour mesurer le délai précis entre l'arrivée de la commande APDU sur l'interface de carte à puce à contact et le départ de la réponse APDU à destination du lecteur (et donc de l'ordinateur).

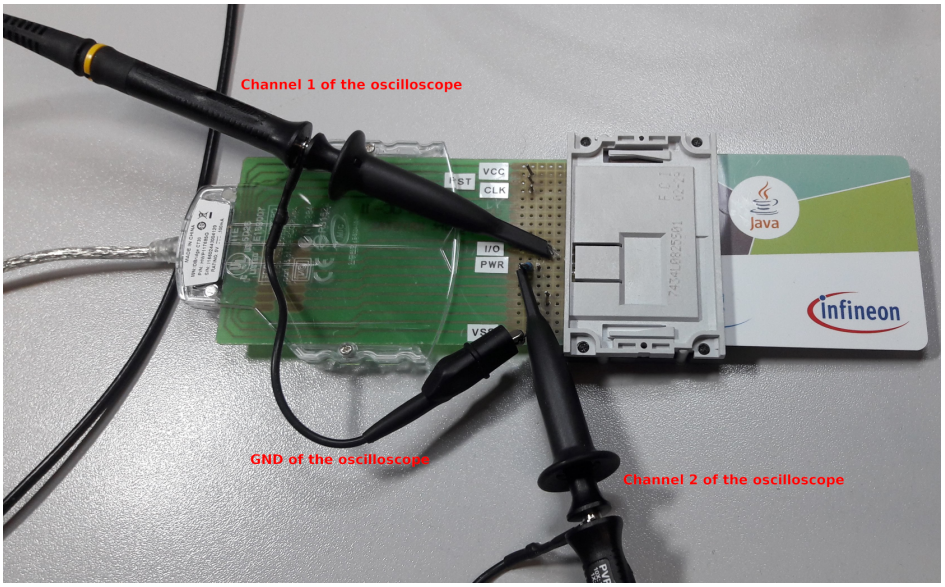


FIGURE 3.8 – Universal JCard exécutant notre applet et connectée au lecteur de carte à puce via un adaptateur permettant des mesures avec notre oscilloscope

Comme illustré sur la Figure 3.8, nous avons connecté le canal 1 de l'oscilloscope sur la ligne I/O et la ligne GND. Nous avons également connecté le canal 2 à une résistance de 22 Ω, qui était placée entre le GND de l'interface de la carte à puce et le GND du lecteur afin de refléter la consommation électrique. Le but de ce canal était de vérifier s'il était possible d'observer un motif évident de consommation d'énergie lors des opérations cryptographiques, afin d'obtenir une meilleure mesure du temps interne consacré à la multiplication scalaire; le motif attendu n'était pas perceptible (probablement en raison de mesures de protection contre les canaux cachés intégrés à l'OS et à la puce).

Temps pour une commande vide

Temps pour une multiplication scalaire sur secp521r1

FIGURE 3.9 – Performances des différentes opérations exécutées sur Universal JCard

Pour éviter tout biais, nous avons effectué les mesures pour chaque courbe elliptique 10 fois. Cela était suffisant car l'écart-type était négligeable et nous avons donc conservé la moyenne. Cependant, dans le code écrit pour effectuer la mesure requise, nous avions quelques instructions supplémentaires, notamment pour obtenir la valeur de P et la copier dans un buffer temporaire en RAM. Il y avait également une petite surcharge sur la carte en raison du dispatcheur APDU du JCRE (« Java Card Runtime Environment »). Aussi pour obtenir les mesures les plus précises de la multiplication scalaire, nous avons souhaité mesurer le temps passé à exécuter toutes ces instructions. Nous avons donc écrit un code spécifique pour les isoler, que nous avons appelé « commande vide » dans la Figure 3.9. Ce code nous a permis de mesurer avec précision ce temps inutile qui est de 5,76 ms (résultat moyen de 10 exécutions).

3.5.2.5 Résultats de performance sur Universal JCard

Les mesures de temps rapportées dans le Tableau 3.12 sont les moyennes pour chaque courbe, à partir desquelles nous avons soustrait le temps utilisé pour la « commande vide ». Nous avions ainsi la meilleure approximation du temps interne passé pour effectuer une multiplication scalaire sur chaque courbe.

Le Tableau 3.12 montre que de manière similaire à Multos IoT Trust Anchor les temps de multiplication scalaire étaient différents pour les paramètres du domaine ayant la même taille en bits. Cependant, pour une même courbe, nous obtenons toujours les mêmes résultats lorsque nous multiplions des scalaires et des points différents en utilisant la multiplication scalaire. Cela est dû aux techniques de programmation en temps constant utilisées par le développeur de l'OS sécurisé afin d'éviter les fuites d'informations. Cela explique également pourquoi seulement 10 exécutions pour chaque courbe étaient suffisantes et pourquoi l'écart-type était négligeable.

Pour rendre les mesures de multiplication scalaires comparables à celles obtenues pour Multos IoT Trust Anchor, nous avons décidé de calculer le temps moyen par taille des paramètres du domaine (c'est-à-dire la taille de la clé privée) et d'illustrer la différence de performances dans la

TABLEAU 3.12 – Temps interne utilisé par la multiplication scalaire sur différentes courbes sous Universal JCard

Taille de la clé privée (bits)	Nom de la courbe	Temps (ms)	Temps moyen (ms)
112	secp112r1	28,04	28,04
128	secp128r1	31,24	31,24
160	secp160k1	38,04	37,89
	secp160r1	37,74	
192	secp192k1	45,74	45,49
	secp192r1	45,24	
224	secp224k1	53,94	53,94
256	Wei25519	59,84	62,06
	secp256k1	63,84	
	secp256r1	62,54	
	brainpoolP256r1	62,04	
384	secp384r1	107,04	107,04
521	secp521r1	196,24	196,24

Figure 3.10.

Les performances pour la multiplication scalaire sur Universal JCard étaient bien meilleures que celles sur Multos IoT Trust Anchor. Il peut y avoir différentes explications, allant de la fréquence interne utilisée (toutefois, seul le SLE78CLUFX5000PHM peut être exécuté à 50 MHz, par rapport à 33 MHz pour le SLE78CLFX4000P) aux implémentations de bibliothèque et des overheads des systèmes d'exploitation. Comme l'objectif de cette thèse ne visait pas à creuser ces différences, nous avons simplement conclu que la multiplication scalaire peut être réalisée de manière efficace et sécurisée sur des dispositifs IoT embarqués.

3.5.2.6 Résumé

Si un niveau de sécurité de 256 bits est requis, et si nous considérons le temps de multiplication scalaire sur Universal JCard, comme indiqué dans le Tableau 3.13, le temps de calcul estimé pour un objet intelligent pour chacune de nos propositions est très raisonnable (dans le calcul les autres opérations ont été considérées comme négligeable au vu des mesures obtenues).

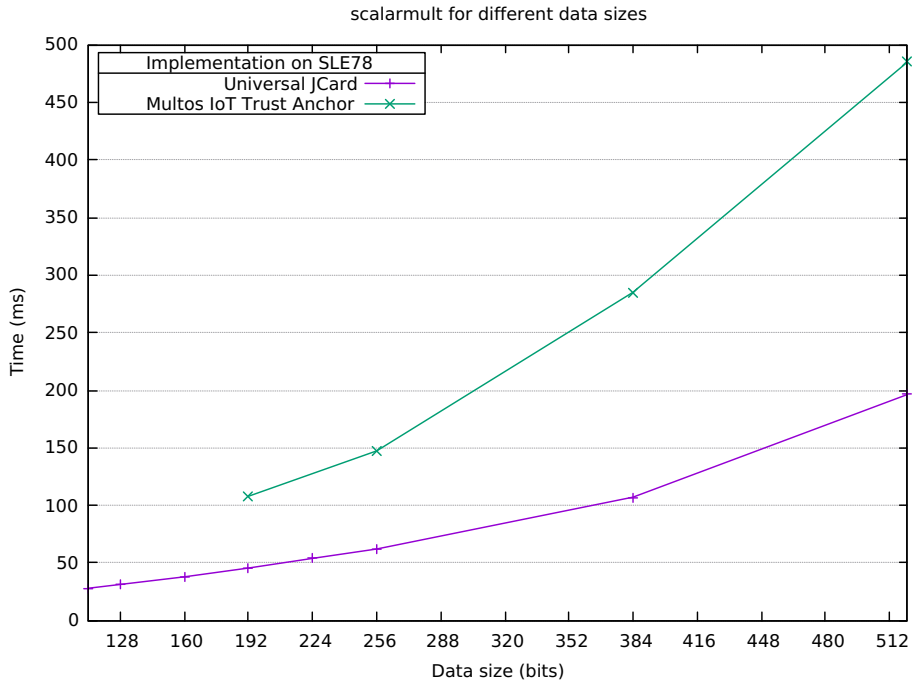


FIGURE 3.10 – Comparaison des performances de la multiplication scalaire exécutée sur les deux cibles à base de SLE78 : Universal JCard et Multos IoT Trust Anchor

TABLEAU 3.13 – Estimation du temps de calcul nécessaire pour l'authentification du côté de l'objet intelligent

Protocole	**Temps estimé (ms)**
Notre amélioration de Jin et al.	186,18
Notre proposition d'authentification basée sur les fonctions de hachage	124,12
Notre protocole d'authentification M2M basé sur ECC	310,30

3.6 Conclusion

Dans ce chapitre, nous avons défini les exigences de sécurité et fonctionnelles des protocoles d'authentification M2C et M2M, puis, nous avons proposé trois protocoles d'authentification mutuelle. Les deux premiers protocoles sont dédiés au mode de communication M2C et le dernier au mode de communication M2M. L'analyse informelle de la sécurité et la vérification formelle à l'aide des outils AVISPA et ProVerif ont montré que nos propositions répondaient à nos exigences de sécurité. De plus, l'analyse des performances et les résultats expérimentaux pratiques ont montré que les protocoles proposés remplissaient également les exigences fonctionnelles énoncées.

En résumé, les trois protocoles d'authentification proposés sont bien plus performants que les

autres propositions disponibles dans la littérature.

Chapitre 4

La proposition d'un protocole d'authentification M2C basé sur le hachage pour les réseaux RFID

Sommaire

4.1 Introduction

Dans ce chapitre nous allons nous intéresser à un autre type de réseaux M2C : les réseaux reposant sur des équipements à faible puissance de calcul et de communication. Plus précisément, nous traiterons de la sécurisation des réseaux RFID (Radio Frequency Identification) pour les applications de santé en nous concentrons sur des solutions légères en calcul (sans courbes elliptiques).

L'identification par radio-fréquences reste une technologie prometteuse permettant la mise en place des systèmes de santé intelligents. En effet, attacher des étiquettes (aussi appelées tag) RFID à différentes entités du système de santé peut fournir une large gamme d'applications comme l'identification et le suivi du personnel hospitalier, la surveillance de l'administration des médicaments pour les patients, le suivi des instruments chirurgicaux, la localisation et la restriction de l'accès aux zones à haut risque des hôpitaux aux personnels autorisés [79, 80, 81]. Pour les patients qui utilisent la technologie RFID, une étiquette RFID peut collecter leurs informations médicales et les transmettre au serveur pour les stocker dans une base de données sous forme de dossier médical électronique (EMR, « Electronic Medical Record »). Dans ce cas, les données de santé des patients sont étroitement liées à leur vie privée. Seules les personnes autorisées par le patient (médecins, soignants, membres de la famille, etc.) doivent pouvoir y accèder via le système RFID. Si un adversaire peut associer l'identifiant de l'utilisateur à son EMR, en plus des identités et des mouvements pouvant être extraits en suivant l'identifiant de l'étiquette, diverses informations personnelles peuvent être acquises à l'insu de leur propriétaire. En outre, un accès non autorisé à un ou plusieurs EMRs peut entraîner la divulgation de données confidentielles à caractère médical. Ainsi, la sécurité et la confidentialité des données des patients sont des préoccupations essentielles qui doivent être traitées avant le déploiement des étiquettes RFID.

Pour garantir la sécurité d'un système médical reposant sur des étiquettes RFID, il est nécessaire d'interdire la transmission de toute information d'identification. On va ainsi parler d'identification anonyme où seules les personnes autorisées peuvent lever l'anonymat et ainsi accéder aux données protégées. Pour cela, une solution privilégiée est l'utilisation de protocoles reposant sur des fonctions de hachage unidirectionnelles sans collisions qui présentent l'avantage de masquer l'information est d'être très légères en terme de puissance de calcul ce qui les rend utilisables sur des étiquettes RFID.

Dans la suite du chapitre, nous commencerons par présenter l'architecture de sécurité considérée au sein de laquelle nous présenterons les hypothèses et les exigences de sécurité que nous avons retenues pour la définition de notre protocole d'identification sécurisé pour les réseaux RFID. Après un état de l'art des propositions existantes, nous présenterons notre proposition pro-

prement dite et terminerons ce chapitre par une preuve de sécurité suivie d'une évaluation de performances de notre proposition. Un article est en cours de rédaction pour une prochaine soumission.

4.2 L'architecture de sécurité considérée

Dans cette section nous présenterons le modèle d'attaque ainsi que les hypothèses et les exigences de sécurités retenues pour la conception de notre protocole M2C. Ce protocole a vocation à être déployé dans un environnement domiciliaire équipé d'étiquettes RFID.

La Figure 4.1 illustre un exemple d'environnement d'assistance médicale RFID comprenant des systèmes RFID d'identification et de surveillance qui collectent et transmettent des données médicales à des serveurs situés dans le cloud. Ces données sont ensuite utilisées pour fournir divers services d'assistance aux personnes à domicile.

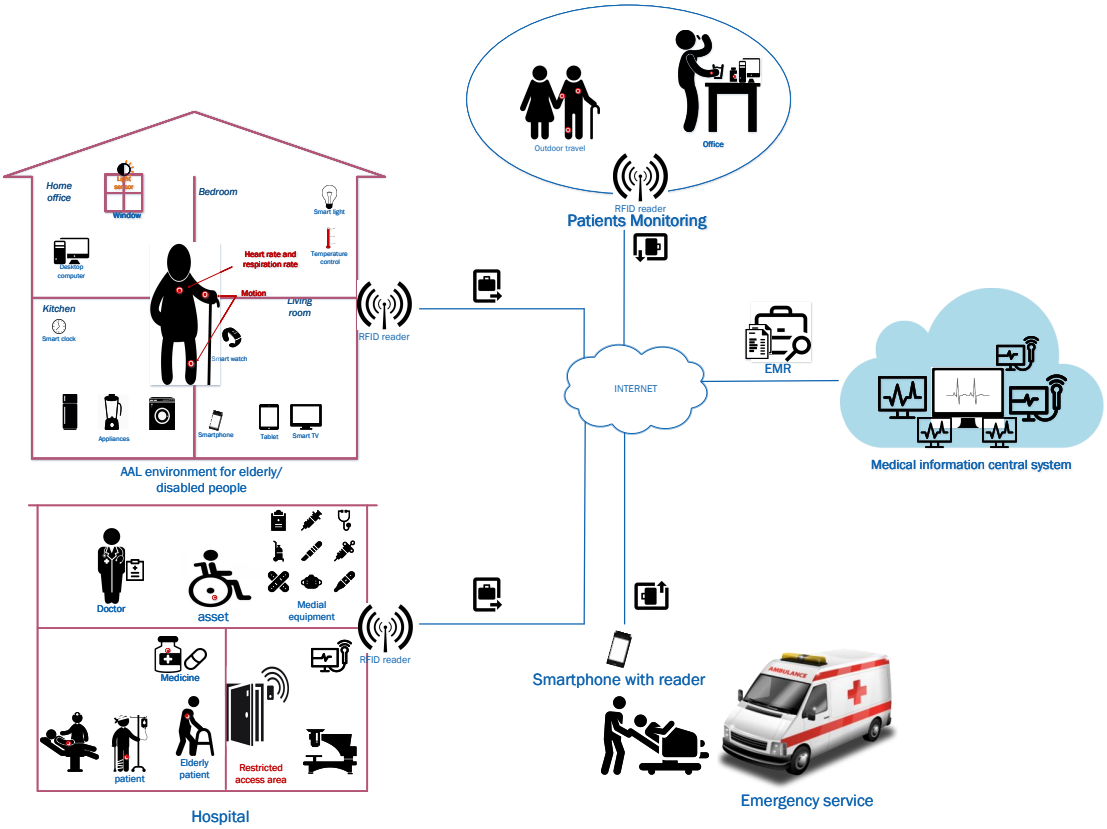


FIGURE 4.1 – Environnement d'assistance médicale utilisant la technologie RFID

4.2.1 Le modèle d'attaque

Le système considéré est constitué de trois entités légitimes : le serveur principal, les lecteurs et les étiquettes.

Les lecteurs RFID sont considérés comme des entités intermédiaires entre le serveur et les étiquettes. Ils diffusent des messages d'interrogation du serveur aux étiquettes et transmettent les réponses des étiquettes correspondantes au serveur sans accéder aux informations secrètes.

En raison des caractéristiques du système RFID, un adversaire peut écouter, intercepter et modifier les messages de communication échangés entre le serveur et les étiquettes.

4.2.2 Les hypothèses de sécurité

Dans notre proposition, les hypothèses sont les suivantes :

- (H1) Le serveur est une entité de confiance avec des ressources illimitées et il ne peut être compromis.
- (H2) Le canal de communication entre le serveur et les étiquettes n'est pas sécurisé et l'adversaire peut lancer une session d'authentification mais ne peut pas compromettre le serveur.
- (H3) Les fonctions de hachage utilisées par le serveur et les étiquettes sont robustes (c'est-à-dire unidirectionnelles, résistantes aux collisions, etc.).

4.2.3 Les exigences de sécurité

Les exigences de sécurité souhaitées pour notre protocole sont répertoriées ci-dessous :

- (ES1) Deux entités communicantes doivent pouvoir s'authentifier mutuellement pour garantir l'identité de l'homologue avec lequel elles communiquent avant d'échanger ou de transmettre des informations.
- (ES2) Un utilisateur non autorisé ne doit pas avoir accès aux informations secrètes échangées pendant la session de communication.
- (ES3) Aucune donnée reçue ne doit être modifiée pendant sa transmission.
- (ES4) Les valeurs secrètes utilisées doivent être mises à jour après chaque établissement de session réussi.
- (ES5) Pour assurer l'anonymat, toutes les informations privées transmises telles que l'identité de l'étiquette et la localisation secrète doivent être protégées.
- (ES6) La confidentialité persistante (perfect forward secrecy) doit être garantie. Il doit être impossible pour un adversaire de récupérer des données secrètes d'une précédente session et ce, même si la session en cours est compromise.

4.2.4 Les notations utilisées

Les notations utilisées pour décrire les différents protocoles d'authentification présentés dans ce chapitre sont décrites dans la Table 4.1.

TABLEAU 4.1 – Notations utilisées dans la description des protocoles

Notation	**Signification**
Tag_i	Numéro de l'étiquette (ou tag) RFID i
ID_{i,j_x}	Identifiant secret de l'étiquette Tag_i numéro j_x, ID_i si Tag_i possède juste un seul identifiant
H_1, H_2, H_3	Trois fonctions de hachage
V_{max}	Valeur maximale du résultat de la fonction de hachage
G_i	Identifiant du groupe i
N	Nombre total d'étiquettes du système
τ	Nombre de groupes dans le système
M	Nombre d'identifiants secrets dans le système
n	Nombre d'étiquettes dans le même groupe
m	Ensemble des identifiants secrets de Tag_i choisis à partir de M, où $m < M$
r_s	Nombre aléatoire du serveur
r_r	Nombre aléatoire du lecteur
r_t	Nombre aléatoire de l'étiquette t
K_{G_i}	Clé secrète du groupe i
K_{Tag_i}	Clé secrète de l'étiquette Tag_i
E_y	Fonction de chiffrement en utilisant la clé y
ξ_i	Ensemble des identifiants secrets du groupe G_i, $\xi_i = \{ID_{i,1}, ID_{i,2}, ..., ID_{i,M}\}$, où $1 \leq i \leq \tau$
Ω_{ij}	Ensemble des identifiants de Tag_i, $\Omega_{ij} = \{ID_{i,j_1}, ID_{i,j_2}, ..., ID_{i,j_m}\}$, où $\{j_x \in \{1,2,...,M\} \mid 1 \leq x \leq m\}$ et $ID_{i,j_x} \neq ID_{i,j_y}$ pour tout $x \neq y$ à partir de l'ensemble des identifiants du groupe ξ_i
π_x	Ensemble des clés secrètes des étiquettes, $\pi_x = \begin{cases} \{k_{\omega_1}, k_{\omega_2}, ...\}, & \omega_* \in \{Tag_1, Tag_2, ..., Tag_N\} \\ \varnothing \end{cases}$
σ_i	Mappage des identifiants de l'ensemble ξ_i avec les clés secrètes des tags, $\sigma_i = \{\langle ID_{i,x}, \pi_x \rangle \mid 1 \leq x \leq M \text{ et } ID_{i,x} \in \xi_i\}$
$r \xleftarrow{\$} \mathbb{Z}_p^*$	Dénote que r est un nombre aléatoire $\in \mathbb{Z}_p^*$
$A \| B$	Dénote l'opération de concaténation des valeurs A et B
$A \stackrel{?}{=} B$	Dénote la vérification de l'égalité entre A et B
$A = B$	Dénote que la variable A reçoit la valeur de l'expression B

4.3 L'état de l'art des protocoles d'authentification RFID

Ces dernières années, de plus en plus de protocoles d'authentification RFID ont été proposés pour résoudre les problèmes de sécurité et de respect de la vie privée qui menacent ces systèmes.

L'un des premier protocole a été proposé par Ohkubo et al. [191] en 2003. Dans ce protocole, les auteurs utilisent un système de hachage en chaîne à faible coût permettant d'identifier en toute sécurité les étiquettes RFID pendant la phase d'identification. L'idée de base de ce protocole est d'utiliser une valeur secrète pour identifier les étiquettes dans le système, puis de l'incrémenter à chaque sollicitation par un lecteur. Ce changement d'identité à chaque échange permet donc d'éviter d'être tracé par un attaquant extérieur ce qui assure d'après les auteurs une confidentialité persistante (perfect forward secrecy). Cependant, Avoine et al. [1] ont analysé le schéma proposé par Ohkubo et al. OSK [191] et ont montré qu'il présentait des limitations importantes en terme de passage à l'échelle. En effet, pour retrouver l'identité de l'étiquette, il est nécessaire de faire une recherche exhaustive dans une base de données ce qui pose des questions de performance lorsque le nombre d'étiquettes est important. Ils ont donc proposé une solution d'optimisation qui sera détaillée dans la section 4.3.1.

En 2006, Tsudik et al. [192] ont proposé d'utiliser l'horodatage (en anglais «timestamping») pour garantir l'authentification anonyme des étiquettes RFID. Ce protocole, appelé YA-TRAP (Yet Another Trivial RFID Authentication Protocol), a été conçu pour identifier des lots d'étiquettes. Cependant, comme l'ont montré Juels et al. [193], YA-TRAP ne peut pas résister aux attaques par déni de service ni garantir l'anonymat des étiquettes.

En 2015, Srivastava et al. [194] ont proposé un protocole d'authentification mutuelle basé sur le hachage pour les systèmes d'information médicaux (Telecare Medicine Information System (TMIS)). Ils ont affirmé que le protocole était résistant aux attaques actives et passives telles que les attaques par falsification, traçabilité et rejeu. Cependant, Li et al. [195] ont montré que ce protocole était vulnérable à la désynchronisation et à l'attaque d'usurpation d'identité du lecteur et que l'intégrité des données échangées ne pouvait être garantie. Ils améliorent l'efficacité de l'authentification en utilisant l'identifiant de l'étiquette. Néanmoins, ils n'ont pas réussi à garantir l'anonymat des étiquettes car l'identifiant de l'étiquette est transmis en clair avec une valeur fixe ce qui entraîne des attaques de traçabilité et de traçabilité arrière.

En 2016, Benssalah et al. [196] ont proposé un double protocole d'authentification RFID-TMIS pour les applications de soins adaptées aux patients situés dans des régions éloignées et à faible densité de population. Le schéma proposé est composé de deux parties, l'une est une amélioration et une extension du protocole de Li et al. [195] pour l'authentification des patients, le second consiste à réaliser des services TMIS. Toujours en 2016, Kumar et al. [197] ont proposé un proto-

cole d'authentification RFID pour les applications de soins de santé dans un environnement Vehicular Cloud Computing (VCC). Ils utilisent la cryptographie à base de courbes elliptiques pour le mécanisme de génération de clé. Toutefois, en raison de la complexité du calcul de la multiplication scalaire, il n'est pas approprié d'utiliser une infrastructure à clé publique sur des étiquettes de faible puissance.

Les schémas d'authentification basés sur le hachage sont donc plus adaptés en raison de leur coût de calcul léger [198]. Alavi et al. [199] ont analysé la propriété du respect de la vie privée de certains protocoles d'authentification RFID existants et ont proposé de nouvelles versions améliorées afin d'en pallier les faiblesses. Cependant, leurs améliorations restent vulnérables aux attaques par rejeu et l'intégrité des données ne peut pas être garantie.

Rahman et al. [2] ont proposé un protocole RFID basé sur le hachage pour résoudre les problèmes du respect de la vie privée dans un système de santé basé sur l'identification par radiofréquence en utilisant l'authentification et en implémentant des techniques de contrôle d'accès. Pour permettre la non-liaison, chaque étiquette se voit attribuer un ensemble d'identifiants. Cependant, les identifiants réellement utilisés dans le message de réponse de l'étiquette ne garantissent pas l'anonymat du système et l'absence de mécanisme de mise à jour des clés conduit à une forward traceability.

En 2018, Wang et al. [3] se sont appuyés sur le protocole de Rahman et al.[2] pour améliorer le protocole d'Alavi et al. [199]. Les auteurs ont proposé un modèle amélioré d'authentification par groupes anonymes afin d'améliorer l'efficacité de l'authentification du serveur principal et de surmonter les attaques par rejeu ainsi que de détecter les données altérées au cours de la phase d'authentification du protocole [199].

Récemment, pour assurer la sécurité des données médicales collectées, Fan et al. [200] ont proposé un schéma léger d'authentification pour les réseaux RFID appliqué au domaine médical. Les auteurs ont affirmé que le protocole pouvait résister aux attaques typiques des systèmes RFID et garantir l'anonymat des étiquettes. Cependant, l'analyse de sécurité montre que le protocole est vulnérable à l'usurpation d'identité du lecteur, aux attaques de traçabilité des étiquettes qui altèrent l'anonymat lors du processus d'authentification et ne permettent pas d'assurer la confidentialité persistante.

Dans la suite de cette section, nous détaillerons les principaux protocoles qui ont servi de base à nos travaux.

4.3.1 Le protocole d'authentification d'Avoine et Oechslin.

En 2005, Avoine et Oechslin [1] ont proposé un protocole d'authentification sécurisé et passant à l'échelle pour les réseaux RFID qui repose sur des fonctions de hachage. Ce protocole est

présenté comme une amélioration du protocole de Ohkubo et al. [191] qui souffrait d'un problème de passage à l'échelle. Ce protocole proposait d'améliorer l'anonymisation (privacy) des étiquettes RFID en modifiant leur identifiant au cours de chaque échange avec un lecteur. Pour cela, il rafraîchissait son identifiant de manière autonome à l'aide de deux fonctions de hachages. Le lecteur de son coté s'appuyait sur une base de données dans laquelle il cherchait l'identité d'origine de l'étiquette qui répondait en appliquant le même algorithme que l'étiquette sur les identités d'origine. La complexité de recherche est donc de $mn/2$ (où m correspond au nombre de sollicitations du lecteur vis à vis du tag et n correspond au nombre de tags) lorsque toutes les étiquettes sont connues à l'avance et ne sont interrogées que par des lecteurs faisant partie du système. Si l'on considère un environnement plus ouvert où les étiquettes peuvent par exemple être sollicitées par d'autres lecteurs, la complexité monte alors à $2mn$.

Pour améliorer cette complexité, Avoine et Oechslin se sont appuyés sur les travaux de Oechslin [201] qui eux-même reposaient sur des travaux originaux de Hellman [202] et qui visaient à réduire le compromis temps/mémoire lors d'une recherche grâce à l'utilisation de fonctions de hachage.

4.3.1.1 La phase de configuration

Les auteurs ont utilisé une variante arc-en-ciel (en anglais «rainbow») [201] de la technique de Hellman qui repose sur différentes fonctions de réduction pour générer plusieurs tables. Ils utilisent alternativement une fonction à sens unique F (4.1) et une fonction de réduction R (4.2) pour construire des chaînes d'entrées et de sorties de F. De plus, ils calculent tous les mappages des étiquettes (ID_i^t, C_i^t) du côté de la base de données.

$$F : (i,t) \mapsto C_i^t = H_1(H_2^{t-1}(ID_i^1)) \tag{4.1}$$

où $1 \leq i \leq n$ et $1 \leq t \leq m$

$$R : C_i^1 \mapsto (i', t') \tag{4.2}$$

où $1 \leq i' \leq n$ et $1 \leq t' \leq m$.
Pour finir, les auteurs stockent le premier et le dernier élément de chaque chaîne dans des tables de la base de données. Chaque Tag_i stocke un identifiant secret ID_i^1 et deux fonctions de hachage H_1 et H_2.

4.3.1.2 La phase d'identification

Cette phase est composée des trois échanges décrits dans la Figure 4.2. De plus, lorsque les étiquettes RFID sont intégrées à des capteurs, les auteurs proposent d'utiliser un canal sécurisé

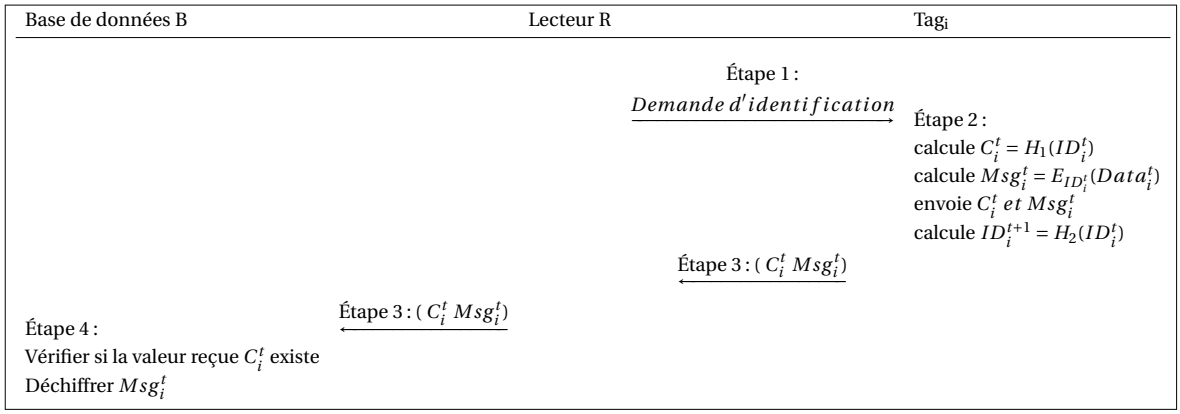


FIGURE 4.2 – Phase d'identification de protocole d'Avoine et al. [1]

supplémentaire pour permettre à l'étiquette Tag_i de transmettre en toute sécurité les données collectées des capteurs au lecteur.

Les étapes suivantes expliquent ce processus d'identification.

- **Étape 1.** Le lecteur envoie une demande d'identification à Tag_i.
- **Étape 2.** Tag_i répond avec $C_i^t = H_1(ID_i^t)$ et chiffre $Data_i^t$ à l'aide de ID_i^t puis génère un nouvel identifiant secret $ID_i^{t+1} = H_2(ID_i^t)$.
- **Étape 3.** Après avoir reçu la réponse de l'étiquette, le lecteur transmet C_i^t et Msg_i^t à la base de données principale.
- **Étape 4.** Pour identifier correctement Tag_i, la base de données B vérifie si C_i^t existe. Si B identifie correctement Tag_i, elle déchiffre le Msg_i^t reçu et extrait $Data_i^t$.

4.3.1.3 Les limitations du protocole

Ce protocole a été proposé pour garantir l'anonymat des étiquettes RFID. Sur la base d'opérations cryptographiques légères, les auteurs ont affirmé qu'il répondait à toutes les exigences de sécurité nécessaires, y compris le respect de la vie privée et la non traçabilité.

4.3.2 Le protocole de Rahman et al.

4.3.2.1 Présentation générale

Le protocole proposé par Rahman et al. se compose de deux éléments principaux : PriSens et HSAC.

- PriSens est un protocole d'authentification privé anonyme et efficace offrant un compromis entre le respect de la vie privée et le passage à l'échelle.

- HSAC est un mécanisme d'accès aux services de santé respectueux de la vie privée basé sur un mécanisme de contrôle d'accès par rôles permettant de limiter l'accès non autorisé aux données. Il repose essentiellement sur deux entités : le gestionnaire de contrôle d'accès (ACM, «Access Control Manager») et le gestionnaire de politiques de vie privée (PPM, «Privacy Policy Manager»).

Dans la suite, nous nous intéresserons principalement à PriSens sur lequel nous avons basé nos travaux. Ce protocole se découpe en deux grandes phases : une phase de configuration (hors ligne) et une phase d'identification (en ligne).

4.3.2.2 La phase de configuration

Pendant la configuration du système, les étiquettes sont divisées en un nombre τ de groupes de taille n. Chaque groupe G_i se voit attribuer un ensemble d'identifiants secrets ξ_i et une clé secrète de groupe K_{G_i}. Chaque étiquette Tag_i appartenant au groupe G_i reçoit une clé secrète K_{Tag_i} et un ensemble d'identifiants Ω_{ij}. De plus, chaque étiquette a au moins un identifiant commun à au moins deux autres membres du même groupe.

Le serveur principal, désigné dans leur article comme étant le lecteur, stocke toutes les informations secrètes des étiquettes et des groupes du système $\psi = \{\langle K_{Gi}, \sigma_i \rangle \mid 1 \leq i \leq \tau\}$. σ_i correspond à l'association de $ID_{i,x}$ aux clés secrètes π_x ou à un ensemble vide $\varnothing$ dans le cas où aucune étiquette n'est associée.

4.3.2.3 La phase d'identification

Lecteur R		Tag_i
Étape 1 : Générer un nombre aléatoire r		
	$\xrightarrow{r}$	Étape 2 : Générer un nombre aléatoire t Choisir un ID_{i,j_x} dans Ω_{ij} $u = E_{K_{G_i}}(r \Vert t \Vert ID_{i,j_x})$ $v = E_{K_{Tag_i}}(r \Vert t)$ $Msg_i = (u, v)$
	$\xleftarrow{Msg_i}$	
Étape 3 : R essaie toutes les clés des groupes jusqu'à déchiffrer u et retrouver ID_{i,j_x} R recherche l'ensemble de clés π_x associé avec ID_{i,j_x} R essaie toutes les clés dans π_x jusqu'à déchiffrer v si une clé existe, R accepte Tag_i, Sinon R rejette Tag_i		

FIGURE 4.3 – Phase d'identification de protocole PriSens [2]

Les étapes suivantes illustrées sur la Figure 4.3 expliquent l'interaction entre le lecteur (R) et une étiquette (Tag_i) au cours de la phase d'identification de PriSens.

- **Étape 1.** Le lecteur R génère un nombre aléatoire r et l'envoie à Tag_i.
- **Étape 2.** En réponse, Tag_i génère d'abord un nombre aléatoire t et choisit un identifiant ID_{i,j_x} dans Ω_{ij}. Par la suite, il calcule $u = E_{K_{G_i}}(r \parallel t \parallel ID_{i,j_x})$ et $v = E_{K_{Tag_i}}(r \parallel t)$.

  Tag_i envoie le message $Msg_i = (u, v)$ à R.
- **Étape 3.** Après la réception de la réponse de l'étiquette, le lecteur déchiffre la première partie de Msg_i en essayant toutes les clés des groupes du système jusqu'à retrouver ID_{i,j_x}.

  Ensuite, R essaie toutes les clés associées à l'identifiant reçu ID_{i,j_x} à partir de l'ensemble π_x pour déchiffrer v.

  Si la clé correspondante est trouvée, R accepte Tag_i, sinon R rejette Tag_i.

4.3.2.4 Les limitations du protocole

Le protocole de Rahman et al. permet donc de prévenir l'association des messages avec une identité grâce à l'ensemble d'identifiants qu'utilise chaque étiquette. Cependant, les clés de chiffrement ne sont pas mises à jour après chaque échange ce qui permet de pouvoir remonter une suite d'échanges précédents en cas de compromission à un moment donné. De plus, le fait de pouvoir remonter les échanges avec une même clé permet en fait de retrouver l'identité de l'étiquette en recomposant l'ensemble de ses identifiants.

En terme de complexité, le protocole de Rahman et al. est linéaire en fonction du nombre de groupes et en fonction du nombre d'identifiants. Plus précisément, sa complexité de recherche est de $\tau + |\pi_x|$ où τ est le nombre de groupes et $|\pi_x|$ est la taille de l'espace de clés. Ainsi, plus le nombre de groupes est important et plus la taille de l'espace de clés augmente, plus le temps de recherche est long.

4.3.3 Le protocole de Wang et al.

En 2018, Wang et al. [3] ont proposé une amélioration du protocole d'authentification de Alavi et al. [199] en utilisant le modèle d'authentification par groupes anonymes introduit par Rahman et al. [2] que nous avons présenté dans la sous-section précédente. Leur protocole améliore l'efficacité de l'authentification du serveur principal et prévient les attaques par rejeu. Il permet également de détecter les données altérées au cours de la phase d'authentification. Pour cela, Wang et al. proposent d'utiliser un horodatage côté lecteur, de chiffrer l'identifiant du lecteur, son nombre aléatoire et une étiquette temporelle (timestamp) et de chiffrer les données des étiquettes envoyées par le serveur au lecteur en s'appuyant sur des fonctions de hachage unidirectionnelles sans collisions. De plus, ils font correspondre les informations des étiquettes avec le message du serveur pour résoudre le problème d'intégrité des données.

Comme les protocoles sur lesquels ils reposent et que nous avons présentés dans les sous-sections précédentes, leur protocole comporte deux phases : une phase de configuration et une phase d'identification.

4.3.3.1 La phase de configuration

Le système configure le générateur de timestamp pour le lecteur et maintient la synchronisation des horloges entre le lecteur R et le serveur S. Le serveur sauvegarde $< K_{G_i}, \theta_{i,j_x}, DATA >$ où $DATA$ représente les données de l'étiquette et $\theta_{i,j_x} = (H(ID_{i,j_x}), ID_{i,j_x}), \pi_x | 1 \leq j_x \leq M, ID_{i,j_x} \in \xi_i$, sachant que :

$$\pi_x = \begin{cases} \{\{k_{\omega_1}, k_{\omega_2}, ...\}, \{k_{\omega_1}^{-1}, k_{\omega_2}^{-1}, ...\}\} & \omega_* \in \{Tag_1, Tag_2, ..., Tag_N\} \\ \varnothing \end{cases}$$

Le serveur enregistre également l'identifiant du lecteur (RID) pour pouvoir l'authentifier ultérieurement. Le lecteur stocke également son identifiant (RID). L'étiquette enregistre l'ensemble des identifiants σ_i, sa clé unique K_{Tag_i} et sa clé de groupe K_{G_i}.

4.3.3.2 La phase d'authentification

Les étapes suivantes, illustrées sur la Figure 4.4, expliquent l'interaction entre le serveur (S), le lecteur (R) et une étiquette (Tag_i) au cours de la phase d'identification. Les fonctions $P(\cdot)$ et $H(\cdot)$ représentent des fonctions de hachages unidirectionnelles sans collisions.

- **Étape 1.** Le lecteur R génère un nombre aléatoire r_r et le transmet à Tag_i.
- **Étape 2.** Après la réception de r_r, Tag_i génère d'abord un nombre aléatoire r_t et choisit un identifiant ID_{i,j_x} dans Ω_{ij}. Puis, il calcule $M_1 = P(K_{G_i} \oplus r_r \oplus r_t) \oplus H(ID_{i,j_x})$, $M_2 = P(r_r \| r_t \| H(ID_{i,j_x}))$ et $M_3 = P(K_{Tag_i} \| r_r \| r_t)$.

  Tag_i envoie le message (M_1 M_2 M_3 r_t) à R.
- **Étape 3.** Le lecteur R enregistre son timestamp T_1 et calcule $M_4 = H(RID \| r_r \| T_1)$. Puis, il envoie le message (M_1 M_2 M_3 M_4 r_r r_t T_1) au serveur S.
- **Étape 4.** Après la réception de la demande d'authentification du lecteur R, le serveur vérifie si $T_2 - T_1 < \triangle T$, T_2 étant son timestamp. Si $T_2 - T_1 < \triangle T$ le serveur continue de vérifier la légitimité du lecteur et de l'étiquette sinon il arrête le protocole.

  Puis, il vérifie si $M_4 \stackrel{?}{=} H(RID \| r_r \| T_1)$. Si l'égalité est vérifiée alors le lecteur est légitime sinon il arrête le protocole. Pour vérifier la légitimité de l'étiquette, S calcule $H(ID_{i,j_x}) = M_1 \oplus P(K_{G_i} \oplus r_R \oplus r_t)$. Puis, il s'assure que $H(ID_{i,j_x})$ est le bon pseudo identifiant choisi par l'étiquette en vérifiant si $M_2 \stackrel{?}{=} P(r_r \| r_t \| H(ID_{i,j_x}))$. Dans le pire des cas, ce processus

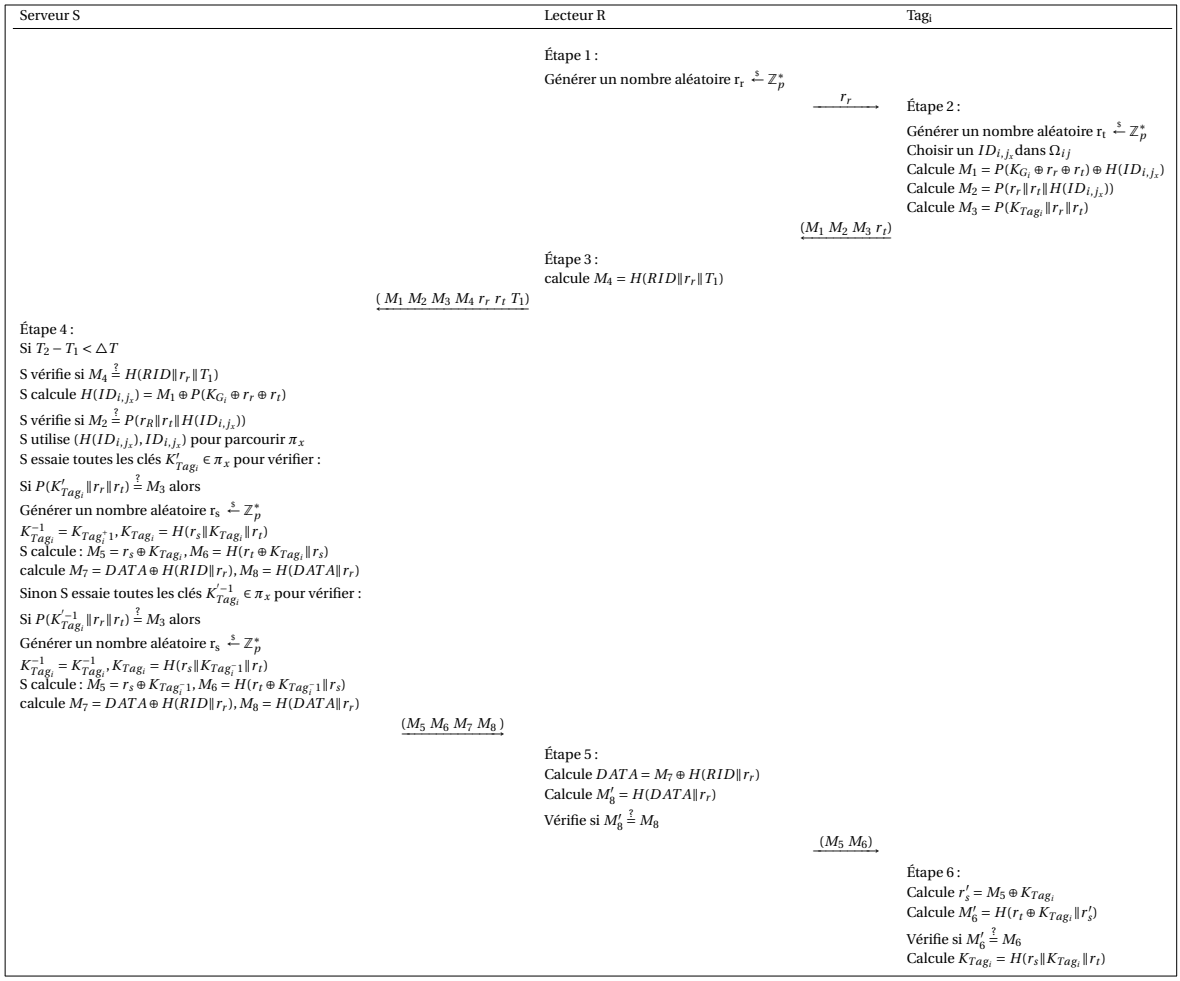


FIGURE 4.4 – Phase d'identification de protocole de Wang et al. [3]

s'effectue τ fois. Une fois $H(ID_{i,j_x})$ retrouvé, S utilise $(H(ID_{i,j_x}), ID_{i,j_x})$ pour parcourir π_x. Par la suite, S essaie toutes les clés $K'_{Tag_i} \in \pi_x$ pour vérifier si $P(K'_{Tag_i} \| r_r \| r_t) \stackrel{?}{=} M_3$. Une fois la clé trouvée, S génère un nombre aléatoire $r_s \xleftarrow{\$} \mathbb{Z}_p^*$ et met à jour les ensembles θ_{i,j_x} qui contiennent les clés $K_{Tag_i}^{-1} = K_{Tag_i}$ et $K_{Tag_i} = H(r_s \| K_{Tag_i} \| r_t)$. S calcule $M_5 = r_s \oplus K_{Tag_i}$, $M_6 = H(r_t \oplus K_{Tag_i} \| r_s)$, $M_7 = DATA \oplus H(RID \| r_r)$, $M_8 = H(DATA \| r_r)$ et envoie le message $(M_5\ M_6\ M_7\ M_8)$ à R.

Si il n'existe pas de K'_{Tag_i} qui satisfasse $P(K'_{Tag_i} \| r_r \| r_t) \stackrel{?}{=} M_3$, S essaie toutes les clés $K'^{-1}_{Tag_i} \in \pi_x$ pour vérifier si $P(K'^{-1}_{Tag_i} \| r_r \| r_t) \stackrel{?}{=} M_3$. S'il trouve la clé, S génère un nombre aléatoire $r_s \xleftarrow{\$} \mathbb{Z}_p^*$ et met à jour les ensembles θ_{i,j_x} qui contiennent les clés $K_{Tag_i}^{-1} = K_{Tag_i}^{-1}$ et $K_{Tag_i} = H(r_s \| K_{Tag_i}^{-1} \| r_t)$. S calcule $M_5 = r_s \oplus K_{Tag_i}$, $M_6 = H(r_t \oplus K_{Tag_i} \| r_s)$, $M_7 = DATA \oplus H(RID \| r_r)$, $M_8 = H(DATA \| r_r)$ et envoie le message $(M_5\ M_6\ M_7\ M_8)$ à R.

- **Étape 5.** Après la réception du message $(M_5\ M_6\ M_7\ M_8)$, R calcule $DATA = M_7 \oplus H(RID \| r_r)$ et vérifie si $M'_8 = H(DATA \| r_r)$ pour s'assurer de l'intégrité de $DATA$.

  R envoie le message $(M_5\ M_6)$ à l'étiquette Tag_i.

- **Étape 6.** Après la réception du message du lecteur, l'étiquette Tag_i utilise sa clé K_{Tag_i} pour extraire r'_s à partir du message M_5 puis calcule $M'_6 = H(r_t \oplus K_{Tag_i} \| r'_s)$. Si l'égalité $M'_6 \stackrel{?}{=} M_6$ est vérifiée, Tag_i authentifie le serveur et met à jour sa clé $K_{Tag_i} = H(r_s \oplus K_{Tag_i} \oplus r_t)$.

4.3.3.3 Les limitations du protocole

Wang et al. ont proposé un protocole de sécurité basé sur l'authentification par groupes anonymes destiné à un environnement RFID où le lecteur est mobile et intervient pendant la communication entre le serveur et l'étiquette. Le serveur doit d'abord authentifier le lecteur avant de pouvoir vérifier l'authenticité de l'étiquette. De la même manière, le lecteur authentifie le serveur avant de vérifier l'intégrité des données de l'étiquette. La principale limitation de ce protocole vient donc paradoxalement de sa principale force, à savoir, l'authentification de toutes les parties impliquées dans la communication qui génère un nombre de messages échangés très important.

La conception de protocoles de sécurité RFID doit prendre en compte les limitations des étiquettes RFID en plus des exigences de l'application. De ce fait, les protocoles de sécurité RFID doivent être à la fois légers et avoir un nombre minimum d'échanges de messages. De notre point de vue, l'authentification du lecteur n'est pas indispensable tant que l'authentification de bout en bout est assurée et que les messages échangés entre le serveur et l'étiquette ne sont pas altérés/-compromis.

4.3.4 Conclusion

Les quelques solutions présentées offrent des bases intéressantes de travail mais ne sont cependant pas suffisantes. En effet, plusieurs de ces solutions présentent soit des failles de sécurité soit des limitations ne leur permettant pas de répondre à l'ensemble des exigences de sécurité que nous nous sommes fixées dans la section 4.2.3. La solution la plus proche de ce que nous cherchons est celle proposée par Wang et al. Cependant, cette solution est complexe et relativement coûteuse en terme de calculs, de nombre d'échanges et de taille de messages échangés.

Dans la suite de ce chapitre, nous allons présenter une nouvelle solution d'authentification légère adaptée aux réseaux RFID qui s'appuie sur la solution initialement proposée par Rahman et al. Ce nouveau protocole repose sur l'utilisation de fonctions de hachage unidirectionnelles sans collisions et sur de l'identification de groupes pour améliorer les temps de recherche et donc les performances générales de la solution proposée.

4.4 La proposition d'un protocole d'authentification RFID basé sur le hachage

Dans cette section, nous décrivons notre proposition protocolaire basée sur les fonctions de hachage unidirectionnelles sans collisions puis nous spécifions la configuration du système ainsi que les paramètres utilisés. Comme dans le cas du protocole de Rahman et al. qui a servi de base de départ à notre réflexion, notre protocole repose sur deux phases : une phase de configuration préalable qui se déroule «hors ligne» et une phase d'identification intervenant «en ligne».

4.4.1 La phase de configuration

Dans le protocole proposé, la phase d'initialisation reste sensiblement la même que celle du protocole PriSens [2]. Néanmoins, le serveur et les étiquettes dans notre protocole stockent trois fonctions de hachage H_1, H_2 et H_3 au lieu de deux. Comme nous le verrons par la suite, ces fonctions de hachages vont nous permettre à la fois de renforcer la sécurité de notre protocole mais également de réduire la complexité du temps de recherche coté serveur.

4.4.2 La phase d'identification

Les étapes suivantes expliquent l'interaction entre le serveur (S) et une étiquette (Tag_i) au cours de la phase d'identification de notre schéma, comme illustré à la Figure 4.5.

- **Étape 1.** Le serveur S génère un nombre aléatoire r_s et le transmet à Tag_i.
- **Étape 2.** En réponse, Tag_i génère d'abord un aléatoire r_t et choisit un identifiant ID_{i,j_x} dans Ω_{ij}. Puis, il calcule $Msg_i = E_{K_{G_i}}(r_s \| r_t \| H_1(K_{Tag_i}) \| ID_{i,j_x})$.

  Tag_i envoie le message Msg_i à S.
- **Étape 3.** Après la réception de la réponse, le serveur déchiffre Msg_i en essayant toutes les clés de groupe dans le système pour retrouver ID_{i,j_x}.

  Ensuite, S utilise $H_1(K_{Tag_i})$ pour retrouver la correspondance K'_{Tag_i} dans la base de données et vérifie si $H_1(K'_{Tag_i}) \stackrel{?}{=} H_1(K_{Tag_i})$. Si une correspondance est trouvée, S accepte Tag_i. Sinon S rejette Tag_i.

  Le serveur calcule $K^{+1}_{Tag_i} = H_2(K_{Tag_i})$ et envoie $H_3(K^{+1}_{Tag_i} \| r_t)$ à Tag_i.
- **Étape 4.** Tag_i reçoit $H_3(K^{+1}_{Tag_i} \| r_t)$ et vérifie si $H_3(K^{+1}_{Tag_i} \| r_t) \stackrel{?}{=} H_3(H_2(K_{Tag_i}) \| r_t)$. Si une correspondance est trouvée, le serveur est authentifié et Tag_i met à jour sa clé suivant $K^{+1}_{Tag_i} = H_2(K_{Tag_i})$.

  Pour éviter les attaques par désynchronisation le serveur enregistre la clé $K^{-1}_{Tag_i}$ de Tag_i.

Serveur S	Lecteur R	Tag_i
Étape 1 :		
Générer un nombre aléatoire $r_s \xleftarrow{\$} \mathbb{Z}_p^*$		
	$\xrightarrow{r_s}$	Étape 2 :
		Générer un nombre aléatoire $r_t \xleftarrow{\$} \mathbb{Z}_p^*$
		Choisir un ID_{i,j_x} dans Ω_{ij}
		Calculer $Msg_i = E_{K_{G_i}}(r_s \Vert r_t \Vert H_1(K_{Tag_i}) \Vert ID_{i,j_x})$
	$\xleftarrow{Msg_i}$	
Étape 3 :		
S déchiffre Msg_i pour retrouver ID_{i,j_x}		
S utilise $H_1(K_{Tag_i})$ pour retrouver		
la correspondance K'_{Tag_i} à partir de π_x		
S vérifie si $H_1(K'_{Tag_i}) \stackrel{?}{=} H_1(K_{Tag_i})$		
Si aucune correspondance n'est trouvée,		
S rejette Tag_i,		
sinon S calcule $K_{Tag_i}^{+1} = H_2(K_{Tag_i})$		
S envoie $H_3(K_{Tag_i}^{+1} \Vert r_t)$		
	$\xrightarrow{H_3(K_{Tag_i}^{+1} \Vert r_t)}$	Étape 4 :
		Vérifier si $H_3(K_{Tag_i}^{+1} \Vert r_t) \stackrel{?}{=} H_3(H_2(K_{Tag_i}) \Vert r_t)$
		Actualiser $K_{Tag_i}^{+1} = H_2(K_{Tag_i})$

FIGURE 4.5 – Phase d'identification de notre protocole

4.4.3 Conclusion

Dans cette section nous avons proposé un nouveau protocole d'authentification reposant sur des fonctions de hachage et pouvant être utilisé par des étiquettes RFID. Ce protocole repart de la proposition faite par Rahman et al. et permet d'en résoudre les principaux problèmes de sécurité ainsi que d'en diminuer la complexité. Dans la suite de ce chapitre, nous présenterons une analyse de sécurité ainsi qu'une analyse de performance de cette proposition.

4.5 L'analyse de sécurité

Dans ce qui suit, nous allons proposer une analyse informelle et une vérification formelle de sécurité de notre protocole d'identification. Ces analyses et vérifications s'appuieront sur les hypothèses et exigences de sécurités présentées en sections 4.2.2 et 4.2.3. Les vérifications de sécurité formelles seront réalisées avec les outils AVISPA [168] et ProVerif [169].

4.5.1 L'analyse informelle de sécurité

Nous commencerons par montrer que notre protocole assure l'authentification mutuelle, la confidentialité, l'anonymat des étiquettes, la disponibilité, la confidentialité persistante, en plus de résister aux attaques de traçabilité et d'usurpation d'identité.

- Authentification mutuelle : Dans le protocole proposé, l'adversaire ne peut pas générer un message légitime Msg_i puisqu'il ne peut pas obtenir la clé secrète de l'étiquette K_{Tag_i} où $Msg_i = E_{K_{G_i}}(r_s \| r_t \| H_1(K_{Tag_i}) \| ID_{i,j_x})$ sous l'hypothèse (A3). De plus, le serveur peut extraire ID_{i,j_x} à partir de sa base de données et authentifier l'étiquette en vérifiant que K_{Tag_i} est présente dans sa base de données.

  L'adversaire ne peut pas construire un message légitime $H_3(K^{+1}_{Tag_i} \| r_t)$ puisqu'il ne peut pas obtenir la nouvelle clé secrète de l'étiquette $K^{+1}_{Tag_i}$ et la valeur aléatoire r_t car, selon les hypothèses (A1,A2), le serveur ne peut être compromis. L'étiquette peut également authentifier le serveur en vérifiant si $H_3(K^{+1}_{Tag_i} \| r_t) \stackrel{?}{=} H_3(H_2(K_{Tag_i}) \| r_t)$. Ainsi, le protocole proposé assure l'authentification mutuelle.

- Confidentialité : À la fin de notre protocole, le serveur S et l'étiquette Tag_i partagent une nouvelle clé de session symétrique $K^{+1}_{Tag_i} = H_2(K_{Tag_i})$. S et Tag_i pourront l'utiliser pour dériver les clés permettant d'assurer la confidentialité, l'intégrité et le respect de la vie privée pendant les échanges ultérieurs.

- Anonymat de l'étiquette : L'étiquette stocke un ensemble d'identités contenant plusieurs identifiants afin d'empêcher toute possibilité de liaison même si l'adversaire écoute la communication. De plus, l'étiquette utilise une fonction de hachage pour générer un pseudo-identifiant en réponse à une requête du serveur. Par conséquent, l'attaquant ne peut connaître que le pseudo-identifiant utilisé dans une réponse et l'identifiant réel ne sera pas révélé. Ainsi, le protocole proposé assure l'anonymat de l'étiquette.

- Disponibilité : La clé secrète de l'étiquette est mise à jour après chaque session d'identification réussie. De plus, après l'exécution du protocole, il n'est pas nécessaire de mettre à jour l'ensemble des identités de l'étiquette. L'étiquette sélectionne une identité pour chaque session d'identification. Par conséquent, le protocole proposé assure la disponibilité.

- Confidentialité persistante : Dans le protocole proposé, même si la clé secrète de l'étiquette K_{Tag_i} est compromise par un adversaire, selon l'hypothèse (A3), la fonction de hachage utilisée est considérée comme étant robuste. Il ne peut donc pas calculer $K^{-1}_{Tag_i}$. Par conséquent, notre protocole assure la confidentialité persistante.

- Résistance à l'attaque de traçabilité : Dans le protocole proposé, l'adversaire ne peut pas tracer l'étiquette même s'il arrive à obtenir son identifiant de session secret ID_{i,j_x}. Pour chaque

nouvelle session, le serveur et l'étiquette génèrent des nouvelles valeurs aléatoires r_s et r_t séparément. L'adversaire ne peut pas suivre l'étiquette sur la base des seuls échanges protocolaires.

- Résistance à l'usurpation d'identité de l'étiquette : Si l'adversaire veut usurper l'identité de l'étiquette auprès du serveur après avoir reçu le message r_s envoyé par le serveur, il doit générer un message légitime $Msg_i = E_{K_{G_i}}(r_s \| r_t \| H_1(K_{Tag_i}) \| ID_{i,j_x})$. Cependant, puisqu'il ne connaît ni l'identifiant de l'étiquette ID_{i,j_x} ni sa clé secrète K_{Tag_i}, il ne peut pas générer la même réponse qu'une étiquette légitime. Ainsi, le protocole proposé résiste à l'usurpation d'identité.

4.5.2 La vérification formelle de sécurité avec AVISPA

Le script HLPSL de notre protocole est présenté dans l'annexe A.5. Il existe deux rôles de base `S` et `T` qui expliquent l'activité du serveur S et de l'étiquette Tag_i.

- Il existe deux agents `T`, `S` qui utilisent trois fonctions de hachage `H1, H2, et H3` et une opération de concaténation `CONCAT`.
- Le secret de la clé secrète de l'étiquette `T`, la clé secrète du groupe et son identité `Kti, Kgi, Idti` sont modélisées à l'aide des prédicats `secret(Kti, sec_kti, {T, S})`, `secret(KGI, sec_kgi, {T, S})`, et `secret(Idti, sec_idt, {T, S}), )` qui sont gérés par le `protocol_id` : `sec_kTi`, `sec_kgi` et `sec_idt` respectivement. Les paramètres `Idti` et `(Rs, Rt)` sont gardés secrets pour `T` et `S`.
- L'authentification mutuelle est réalisée via les objectifs `witness` et `request`, c'est-à-dire `witness(T, S, auth_kti, Kti)`, `request(S, T, auth_kti, Kti)`, `witness(S, T, auth_kti, Kti')`, `request (T, S, auth_kti, Kti'). witness(T, S, auth_kti, Kti')` déclare que l'agent `T` prétend être l'homologue de l'agent `S`, en accord sur la valeur `Kti'`. `auth_Kti` représente l'authentification de `Kti'` indiquée dans la section `goal` alors que `request(S, T, auth_kti, Kti')` déclare que l'agent `S` accepte la valeur `Kti'` et s'appuie désormais sur la garantie que l'agent `T` existe et l'accepte pour cette valeur.
- L'environnement contient les constantes globales et la composition d'une ou plusieurs sessions. L'intrus participe en tant que session concrète au protocole d'exécution.

Les résultats après l'exécution de notre protocole codé en HLPSL sont donnés dans le Tableau 4.2.

Comme le montre le Tableau 4.2, AVISPA retourne SAFE à partir de ses deux back-ends On-the-Fly Model-Checker (OFMC) et CL-based Model-Checker (CL-AtSe). Par contre, SAT-based Model-Checker (SATMC) et Tree Automata based on Automatic Approximations for the Analysis of Secu-

TABLEAU 4.2 – Résultats de validation AVISPA

Moteur AVISPA	Résultat
$OFMC$	SAFE
$CL-AtSe$	SAFE
$SATMC$	INCONCLUSIVE
$TA4SP$	INCONCLUSIVE

rity Protocols (TA4SP) retournent INCONCLUSIVE en raison d'opérations non supportées. Ce qui signifie que AVISPA ne peut pas trouver des attaques contre notre protocole.

4.5.3 La vérification formelle de sécurité avec ProVerif

Dans les paragraphes suivants, les points importants de la spécification du protocole et les résultats sont mis en évidence.

Les scripts Pi Calculus appliqués sont présentés dans l'Annexe B.8.

- Les secrets `idt, kti, kgi` sont déclarés comme secret pour l'attaquant en utilisant le mot `[private]`. `ch` est le canal de communication publique où `SERVERS` et `TAGI` échangent leur messages et `h1, h2, h3, concat, concat2` et `enc` représentent des fonctions de hachage, concaténation et de chiffrement respectivement.
- Le secret de `kti`, `kgi` and `idt` est vérifié avec requête `query attacker(kti), query attacker(kgi)` et `query attacker(idt)`.
- L'authentification mutuelle entre le serveur et l'étiquette Tag_i est modélisée par la définition de quatre événements "mappés" dans les sous-processus de `TAGI` et `SERVERS` et les requêtes suivantes

```
event beginT( bitstring ).
event endT( bitstring ).
event beginS( bitstring ).
event endS( bitstring ).
query x: bitstring ; inj -event(beginS(x)) ==> inj -event(beginT(x )).
query x: bitstring ; inj -event(endT(x)) ==> inj -event(endS(x )).
query x: bitstring , y: bitstring ; inj -event(endS(y)) ==>
inj -event(beginS(x)) && inj -event(beginT(x )).
```

- Dans le processus principal, les sous-processus `TAGI` et `SERVERS` sont exécutés en parallèle. `!` indique un nombre de processus illimité :

```
process
((!SERVERS) | (!TAGI))
```

Les résultats disponibles dans l'Annexe B.9 montrent que la sécurité de `idt, hidt` est préservée par le protocole et que l'authentification mutuelle entre `SERVERS` et `TAGI` est bien réalisée.

4.6 L'analyse des performances

Dans cette section nous allons réaliser une analyse de performances de notre protocole à travers une étude de la complexité de notre proposition par rapport aux protocoles de Rahman et al. et de Wang et al. Nous avons souhaité nous comparer à Rahman et al. car il a servi de base à notre proposition et à Wang et al. car, en étant parti de la même base, ils sont arrivés à une solution sécurisée relativement différente. Nous montrerons notamment que notre solution présente de meilleures performances en terme de complexité que la solution de Wang et al. Nous montrerons également que notre protocole présente de meilleurs coûts de communication.

Dans ce cadre, nous considérons un système composé de N étiquettes RFID réparties en τ groupes. n représente le nombre d'étiquettes appartenant au même groupe. T_C et T_H désignent les temps que prennent respectivement une fonction de chiffrement et une fonction de hachage.

4.6.1 L'analyse de complexité

Dans notre proposition, la communication se déroule entre le serveur et l'étiquette contrairement au protocole de Rahman et al. [2] et Wang et al. [3]. Dans [2], la communication implique seulement le lecteur et le tag RFID. Dans ce cas, le lecteur doit sauvegarder une liste d'accès qui est une liste de n tags autorisés à s'authentifier, ce qui fait que le lecteur doit avoir une capacité mémoire de stockage plus élevée. Le protocole de Wang et al. implique les trois entités à savoir le serveur S, le lecteur R et l'étiquette Tag_i.

Dans notre proposition, après la réception de la réponse de l'étiquette Tag_i, le serveur S doit rechercher la clé du groupe correspondante afin de déchiffrer le message reçu. Dans le pire des cas, il effectue cette opération τ fois où τ correspond au nombre de groupes dans le système. Si la clé du groupe existe, le serveur déchiffre le message reçu et extrait $H_1(K_{Tag_i})$. À cette étape, comme dans le protocole de Wang et al. et contrairement au protocole de Rahman et al., le serveur n'effectue aucune recherche. En effet, comme l'avaient proposé Avoine et Oechslin, le serveur S utilise $H_1(K_{Tag_i})$ comme clé d'entrée de la table de hachage de la base de données, ce qui permet d'avoir un accès direct à la clé K_{Tag_i} associée.

La complexité mémoire de notre protocole a légèrement augmentée par rapport au protocole de Rahman et al. mais reste moins élevée que celle de Wang et al. où le serveur a besoin de sto-

cker les données d'identification du lecteur en plus de celle des tags. Dans notre protocole, les étiquettes RFID ont besoin de stocker un ensemble d'identifiants, la clé de groupe, la clé secrète K_{Tag_i}, d'implémenter trois fonctions de hachage H_1, H_2 et H_3. Néanmoins, le protocole proposé s'exécute plus rapidement du côté du serveur.

Les coût de calculs pour le serveur, le lecteur et les étiquettes lors de la phase d'authentification de notre protocole et ceux du protocole de Wang et al. sont respectivement résumés dans le Tableau 4.3.

Protocole	**Coût de calculs**		
	Serveur	**Lecteur**	**Étiquette**
Wang et al.	$2\tau T_C + 5T_H$	$3T_H$	$3T_C + 4T_H$
Rahman et al.	0	$\tau T_C + nT_C$	$2T_C$
Notre protocole	$\tau T_C + 2T_H$	0	$1T_C + 3T_H$

TABLEAU 4.3 – Étude de complexité

Le Tableau 4.3 montre que notre protocole est meilleur en termes de coûts de calcul pour le serveur, le lecteur et l'étiquette RFID. Ceci s'explique principalement par le fait que notre lecteur sert de relais entre le serveur et les étiquettes RFID et n'intervient pas pendant la phase d'authentification. La contrepartie est bien sûr que ce lecteur n'est pas authentifié vis à vis de l'étiquette ou du serveur contrairement au protocole de Wang et al. Par rapport au protocole de Rahman et al., il faut également souligner que nous en corrigeons les limitations de sécurité et ce, pour un moindre coût de calcul.

4.6.2 Les coûts de communication

Dans cette section, nous avons calculé la taille totale en bits des messages transmis au cours de la phase d'authentification et nous avons également compté le nombre d'échanges entre les différentes entités du système. Pour simplifier l'analyse, nous avons considéré que le résultat de la fonction de hachage est de 256 bits et que le résultat de l'horodatage est de 16 bits. Les autres données sont fixées à 128 bits et le résultat de la fonction de chiffrement correspond à la taille des données en clair fournies en paramètre.

Le Tableau 4.4 montre que notre protocole est meilleur en termes de coût de communication entre le serveur et les étiquettes RFID que celui de Wang et al. Là encore cela peut s'expliquer par le choix que nous avons fait de ne pas authentifier le lecteur entre le serveur et les étiquettes dans la mesure où il n'intervient pas dans la communication et n'est utilisé que comme un relais. Les résultats montrent de moins bon résultats que ceux de Rahman et al. mais il faut souligner que

Protocole	Serveur-Lecteur (bits)	Lecteur-Étiquette (bits)	Serveur-Étiquette (bits)	# d'échanges
Wang et al.	2576	1792	0	5
Rahman et al.	0	768	0	2
Notre protocole	0	0	1024	3

TABLEAU 4.4 – Étude des coûts de communication

nous corrigeons des failles de sécurités par rapport à leur proposition.

4.7 Conclusion

Dans ce chapitre nous nous sommes intéressés à un type de réseau M2C particulier s'appuyant sur la technologie RFID. En effet, l'identification par radio-fréquences est une des principales technologies utilisée dans l'Internet des Objets en général et dans son application au domaine médical en particulier. Cette technologie permet la transmission de données sensibles concernant à la fois l'identité mais également des informations sensibles du point de vue de la santé de la personne surveillée. Dans ce cadre, nous nous sommes intéressés aux solutions d'identification sécurisées basées sur des fonctions de hachages unidirectionnelles sans collisions. Ces solutions ont l'avantage d'être relativement légères en terme de puissance de calcul nécessaire en comparaison des solutions basées sur les courbes elliptiques vues dans le chapitre précédent.

Après avoir présenté les hypothèses et exigences de sécurité considérées, nous avons fait un rapide panorama de solutions de sécurité existantes basées sur les fonctions de hachage et adaptées aux réseaux RFID. Nous avons ensuite détaillé trois protocoles proposés respectivement par Avoine et Oechslin, par Rahman et al. et par Wang et al. qui servent de références et de base de comparaison à nos propre travaux. La proposition proprement dite est ensuite détaillée et nous montrons que ce nouveau protocole d'identification léger permet de garantir le respect de la vie privée et la confidentialité des étiquettes RFID via une identification sécurisée reposant sur de l'identification de groupes. L'anonymat des étiquettes est ainsi conservé pendant les échanges et les entités participantes à la communication peuvent réaliser une authentification mutuelle (suivant une approche M2C). Nous avons ensuite réalisé deux analyses de sécurité pour valider notre proposition. Une analyse de sécurité informelle a tout d'abord été menée, suivie d'une vérification formelle réalisée à l'aide des outils AVISPA et ProVerif, démontrant ainsi que notre protocole est sécurisé et répond à nos exigences de sécurité. En conclusion du chapitre, une analyse de performances prenant en compte la complexité et les coûts de communication vient montrer l'intérêt de cette proposition.

Chapitre 5

La proposition d'un protocole de communication sécurisée d'assistance médicale domiciliaire

Sommaire

5.1 Introduction

Avec l'évolution de l'Internet des Objets et son application au domaine médical, la perception de la médecine a changé et elle s'accompagne de l'apparition de nouvelles applications. L'une d'elles est la surveillance à distance des patients depuis leur domicile; ce qui est un vrai besoin. En effet, avec l'évolution démographique de la population et l'augmentation de l'espérance de vie des personnes âgées, le système de santé traditionnel doit se réinventer. Cela s'accompagne à la fois d'une demande de plus en plus importante et légitime des patients d'avoir des informations plus complètes sur leur état de santé mais aussi d'un désir de ces mêmes patients d'avoir une plus grande autonomie dans la prise en charge de leur santé au quotidien.

L'utilisation des dispositifs intelligents par les patients, que ce soit des dispositifs implantés, portés ou à proximité du corps, permet d'obtenir des données physiologiques sur leur état de santé. Ces différents dispositifs peuvent être regroupés sous un même réseau BAN (voir section 1.5.2). Un réseau BAN se compose typiquement d'un certain nombre de nœuds et d'un contrôleur BAN. Ce dernier peut être représenté par le BCU (« Body Control Unit ») qui sert alors de passerelle pour transmettre les données vers un point d'accès distant comme un serveur médical ou le cloud [203].

Comme la procédure de surveillance médicale des patients se base sur la collecte de plusieurs paramètres qui sont soit relatifs au corps humain, soit à l'environnement qui entoure le patient [204], la sécurité des données collectées dans un réseau BAN est une préoccupation majeure tant elles sont en lien direct avec la vie privée de la personne surveillée. Ainsi, pour préserver la sécurité des données personnelles du patient et garantir le respect de sa vie privée, il est nécessaire d'implémenter des mécanismes de sécurité efficaces et adaptés aux ressources de l'environnement BAN. En particulier, ils devront contrer les attaques comme l'usurpation d'identité, la manipulation et l'interception des échanges entre le BCU et les dispositifs intelligents.

Ainsi, dans ce chapitre, nous allons présenter un travail en cours, initié en collaboration avec Olivier Blazy, enseignant-chercheur à l'université de Limoges, sur un protocole de communication sécurisée entre les dispositifs connectés du patient et le BCU afin de permettre la collecte de données. Ce protocole a été inspiré par celui proposé par Won et al. [205] entre un drone et des objets intelligents mais il l'améliore grandement du point de vue de la sécurité des données (en particulier collectées) tout en conservant une faible complexité calculatoire pour les entités communiquantes. Une première implémentation de ce protocole a été réalisée sur TelosB en utilisant TinyOS et le langage NesC. Un article est en cours de rédaction pour une prochaine soumission.

5.2 La sécurité et le respect de la vie privée dans les réseaux WBAN

L'utilisation des réseaux BAN par les patients a pour principal objectif de fournir un accès instantané à leurs données de santé aux entités autorisées comme le médecin traitant par exemple. Assurer la sécurité des données collectées auprès des nœuds consiste à les protéger contre tout accès non autorisé par des entités tierces, c'est-à-dire pendant les différentes opérations de transfert, de traitement et de stockage. Afin d'assurer le respect de la vie privée, seule l'autorité ayant le droit d'accéder à ces informations personnelles doit y être autorisée. Au delà de la confidentialité qui est souvent la propriété la plus considérée, il est également important d'assurer l'intégrité des données car de fausses informations pourraient être transmises au médecin, ce qui pourrait entraîner des conséquences critiques [206, 207, 208, 209]. Dernièrement, plusieurs études ont été menées sur l'importance de l'authentification pour garantir la sécurité des données collectées sur les réseaux WBAN. Zhang et al. [210] ont présenté un schéma de sécurité qui permet aux nœuds dans les BAN de partager une clé commune générée par des signaux d'électrocardiogramme (ECG). Cependant, les schémas basés sur les signaux physiologiques reposent sur l'hypothèse que les différents capteurs portés peuvent mesurer le même type de paramètres ECG ce qui limite la portée d'application de cette approche. En outre, ce type de schéma peut être vulnérable à une attaque par déni de service car il peut y avoir une différence dans les signaux physiologiques d'une même personne, mesurés par des dispositifs différents. De ce fait, Li et al. [211] ont proposé un schéma d'authentification mutuelle anonyme centralisé pour permettre aux nœuds capteurs attachés au corps du patient de s'authentifier auprès d'un serveur local (nœud hub) et d'établir une clé de session de manière anonyme. Toutefois, cette solution n'assure pas la sécurité entre le serveur local et le cloud et n'offre donc pas de sécurité de bout en bout. Dans [212], les auteurs proposent un nouveau schéma d'authentification léger adapté au déploiement de dispositifs portables. Le schéma permet à un utilisateur d'authentifier ses dispositifs et son terminal mobile et d'établir une clé de session pour chaque dispositif afin de garantir une communication sécurisée. Cependant, leur algorithme ne considère pas la sécurité entre le terminal mobile et le cloud [213]. De plus, Kim et al. [214] démontrent que ce schéma est vulnérable aux attaques d'usurpation d'identité, de divulgation de clés de session et de vol de dispositifs. Pour contrer ces menaces, ils proposent un système d'authentification mutuelle et d'établissement de clés, mais ne vont pas plus loin concernant la collecte des données.

Won et al. [205] ont proposé un protocole de communication sécurisée pour protéger les données récoltées à partir des objets intelligents. Appliqués au contexte qui nous intéresse, ces travaux entre un drone et des objets intelligents se transposent à un BCU et des objets intelligents. Toutefois, à nouveau, ce protocole ne garantit pas une sécurité de bout en bout (entre les objets intel-

ligents et le cloud), puisque les données des objets intelligents sont chiffrées par la clé de session avant d'être transférées au BCU qui les stocke. Or, même si pour éviter l'accès aux données par des tierces personnes, les clés privées émises ne sont valides que pour des périodes spécifiques dont la durée peut être courte (une fois la période terminée, de nouvelles clés privées sont générées), le risque de fuite d'informations en cas de capture physique du BCU est important.

Toutefois, ces travaux étant très intéressants, nous nous en sommes inspirés pour proposer une solution assurant une sécurité de bout en bout, même en cas de capture physique du BCU. Nous avons aussi amélioré le protocole contre une attaque possible de déni de service.

Tout naturellement, ce protocole sera notre référence pour la comparaison avec nos travaux.

5.3 Le modèle d'une architecture d'un environnement WBAN

Dans cette section nous présenterons un exemple d'architecture d'un environnement WBAN dans une maison intelligente.

Les différents objets intelligents présents dans l'environnement de l'utilisateur et les dispositifs portés par ce dernier communiquent avec le BCU qui peut être le téléphone portable (smartphone) de l'utilisateur comme illustré Figure 5.1. Dans un contexte médical, afin de les transmettre pour une analyse ultérieure, ces dispositifs sont capables de collecter des données des patients telles que l'activité cérébrale (EEG), la température, la fréquence respiratoire, la glycémie, la prise de médicaments, la saturation en oxygène dans le sang, la fréquence cardiaque, les déplacements et bien d'autres mesures utiles.

Outre la collecte d'informations vitales très détaillées sur l'état de santé, l'utilisation des WBAN dans le cadre de l'AAD permet aussi aux utilisateurs d'avoir une surveillance complète de leur bien-être. Ainsi, les personnes âgées qui ne disposent pas des moyens nécessaires pour vivre dans un établissement d'hébergement pour personnes âgées dépendantes (EHPAD) ou celles qui désirent garder une certaine autonomie bénéficient d'une assistance permanente à domicile. Pour assurer la sécurité de ces personnes, il est fréquent d'enregistrer également leurs mouvements et leurs positions afin de notifier un proche si la personne s'éloigne trop longtemps de zones préalablement déterminées.

Les patients atteints de maladies chroniques peuvent aussi bénéficier de l'utilisation des WBAN afin de compléter leur dossier médical avec des détails qui peuvent ne pas apparaître durant la consultation. Par exemple, le WBAN peut permettre la détection d'états anormaux apparaissant après la prise de certains médicaments et ainsi aider à préciser les diagnostics médicaux du patient.

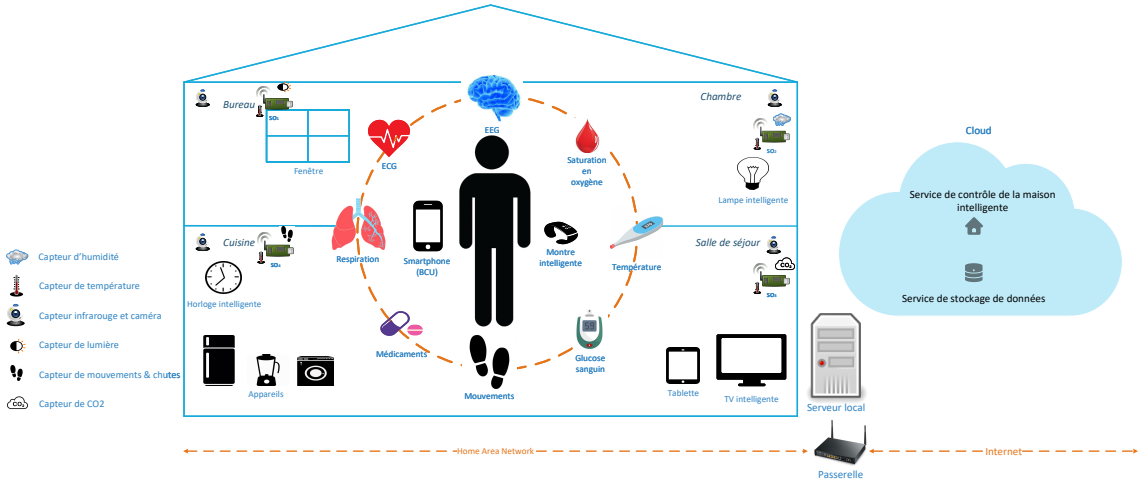


FIGURE 5.1 – Architecture d'un environnement WBAN dans une maison intelligente

5.4 La présentation du protocole de communication sécurisée entre le BCU et les objets intelligents pour la collecte des données

Dans cette section, nous allons présenter notre protocole de communication sécurisée entre le BCU et les objets intelligents (SO) pour la collecte des données. Afin de pouvoir le comparer à celui de Won et al. [205] qui nous a inspiré, nous avons gardé dans le protocole différents algorithmes (qu'ils proposaient mais que nous modifions), à savoir Setup, SetSecretValue, KeyRegistration, SetPrivateKey, SetPublicKey, SymmetricKeyGen, qui sont exécutés par le BCU et les SO afin de s'authentifier mutuellement et d'échanger des données.

5.4.1 Setup

Cet algorithme est exécuté par le Centre de Génération des Clés (CGC) pour fixer les paramètres du système (q, a, b, p, P) à utiliser pour la courbe elliptique, E, pour choisir la fonction de hachage H et pour générer sa paire de clés publique/privé. Le CGC choisit une valeur aléatoire $x_{CGC} \xleftarrow{\$} \mathbb{Z}_p^*$ qui est sa clé privée et calcule sa clé publique comme suit : $P_{CGC} = x_{CGC} \cdot P$. Pour finir, CGC partage avec les autres entités (c'est-à-dire le BCU et les SO) les paramètres du système, son identité et sa clé publique.

5.4.2 SetSecretValue

Cet algorithme permet de générer une valeur secrète et la valeur publique correspondante pour chaque entité du système. Si l'on note A, une entité du système, et ID_A, son identité, A choisit une valeur aléatoire $x_A \xleftarrow{\$} \mathbb{Z}_p^*$, sa valeur secrète (qui deviendra sa clé privée pour nous dans l'algorithme SetPrivateKey – ce n'est pas le cas pour Won et al. [205]) et calcule la valeur publique

correspondante $P_A = x_A \cdot P$.

5.4.3 KeyRegistration

L'entité A demande à l'autorité ID_{CGC} d'authentifier sa valeur publique P_A (c'est-à-dire de la signer – ou dit autrement de la certifier). Pour cela, A envoie au CGC son identité ID_A et sa valeur publique P_A.

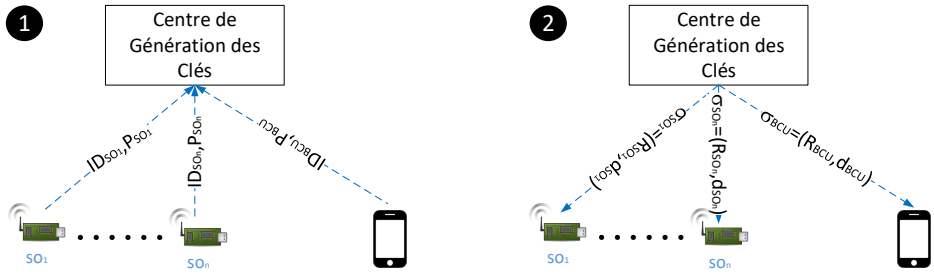


FIGURE 5.2 – Phase de génération et de distribution de clés par le CGC

Le CGC génère d'abord une valeur aléatoire $r_A \xleftarrow{\$} \mathbb{Z}_p^*$ puis calcule :

- $R_A = r_A \cdot P$, un point
- $d_A = r_A + H(ID_{CGC}, ID_A, R_A, P_A, t_A) \cdot x_{CGC}$, une signature de Schnorr pour certifier P_A, où t_A est la date d'expiration de la signature.

Comme illustré Figure 5.2, le CGC renvoie ensuite $\sigma_A = (R_A, d_A)$ pour chaque entité. Donc :

❶ Les objets intelligents $\{SO_1, ..., SO_n\}$ et BCU envoient leurs identités et leurs valeurs publiques au CGC.

❷ Le CGC calcule et envoie le certificat σ_A correspondant pour chaque entité.

Une entité A peut vérifier si le $\sigma_A = (R_A, d_A)$ reçu est valide en effectuant le calcul : $d_A \cdot P = R_A + H(ID_{CGC}, ID_A, R_A, P_A, t_A) \cdot P_{CGC}$, sachant que ID_{CGC} tout comme P_{CGC} sont publiques.

5.4.4 SetPrivateKey

Dans notre proposition, cet algorithme est minimal et consiste en une affectation. Il est exécuté par chaque entité A pour obtenir sa clé privée.

$sk_A = x_A$

5.4.5 SetPublicKey

Cet algorithme est exécuté par chaque entité A pour générer sa clé publique complète.

$pk_A = (P_A, \sigma_A, ID_A, t_A)$.

On notera que toutes ces valeurs sont connues du CGC.

5.4.6 SymmetricKeyGen

- Le BCU envoie ses informations publiques $pk_{BCU} = (P_{BCU}, \sigma_{BCU}, ID_{BCU}, t_{BCU})$ et les authentifie par signature de Schnorr :
  - Il choisit une valeur aléatoire $s_{BCU} \xleftarrow{\$} \mathbb{Z}_p^*$ et calcule $S_{BCU} = s_{BCU} \cdot P$
  - Il calcule $e_{BCU} = s_{BCU} + H(pk_{BCU}, T_{BCU}) \cdot x_{BCU}$ où T_{BCU} est le temps actuel
  - Il définit et envoie $\hat{\sigma}_{BCU} = (S_{BCU}, e_{BCU}, T_{BCU})$
- L'objet intelligent SO vérifie tout d'abord pour des raisons de performances que T_{BCU} correspond au temps actuel et que $T_{BCU} \leq t_{BCU}$ puis il vérifie la validité de σ_{BCU} et $\hat{\sigma}_{BCU}$.
- Il choisit $s_{SO} \xleftarrow{\$} \mathbb{Z}_p^*$ et calcule :
  - $S_{SO} = s_{SO} \cdot P$
  - $K_{SO} = s_{SO} \cdot P_{CGC}$, la clé de chiffrement des données
  - $C_{K_{SO}}(M) = K_{SO} \oplus M$, où $C_{K_{SO}}(M)$ est le chiffré de M (les données) en utilisant le masque jetable K_{SO}
  - $e_{SO} = s_{SO} + H(pk_{SO}, C_{K_{SO}}(M), T_{SO}) \cdot x_{SO}$, où T_{SO} est le temps actuel
  - $\hat{\sigma}_{SO} = (S_{SO}, e_{SO}, T_{SO})$

  Puis le SO envoie ses informations publiques, les données chiffrées et leur authentification par la signature de Schnorr, soit respectivement $pk_{SO}, C_{K_{SO}}(M), \hat{\sigma}_{SO}$.
- Toujours pour des raisons de performance, le BCU vérifie tout d'abord si T_{SO} correspond au temps actuel et que $T_{SO} \leq t_{SO}$ sont valides puis, il vérifie enfin la validité de σ_{SO} et $\hat{\sigma}_{SO}$ avant d'enregistrer la valeur chiffrée $C_{K_{SO}}(M)$ et S_{SO} en mémoire.
- Dans une étape ultérieure, pour déchiffrer les données, le CGC utilisera x_{CGC} pour retrouver K_{SO} en calculant $K_{SO} = x_{CGC} \cdot S_{SO}$ $(= x_{CGC} \cdot s_{SO} \cdot P = s_{SO} \cdot P_{CGC})$ puis il effectuera l'opération inverse $M = K_{SO} \oplus C_{K_{SO}}(M)$.

5.4.7 La collecte des données en utilisant le protocole en mode LPL

Pour illustrer la collecte des données en utilisant notre protocole en mode écoute à faible consommation, soit LPL («Low Power Listening ») [215], nous avons positionné sur la Figure 5.3 le BCU et deux SO.

Étant donné que les objets intelligents et le BCU disposent de ressources limitées, leur énergie doit être utilisée efficacement. Aussi, en mode LPL, notre protocole utilise deux canaux de communication pour le BCU, un canal de réveil et un canal de données et seulement une écoute LPL pour les objets intelligents.

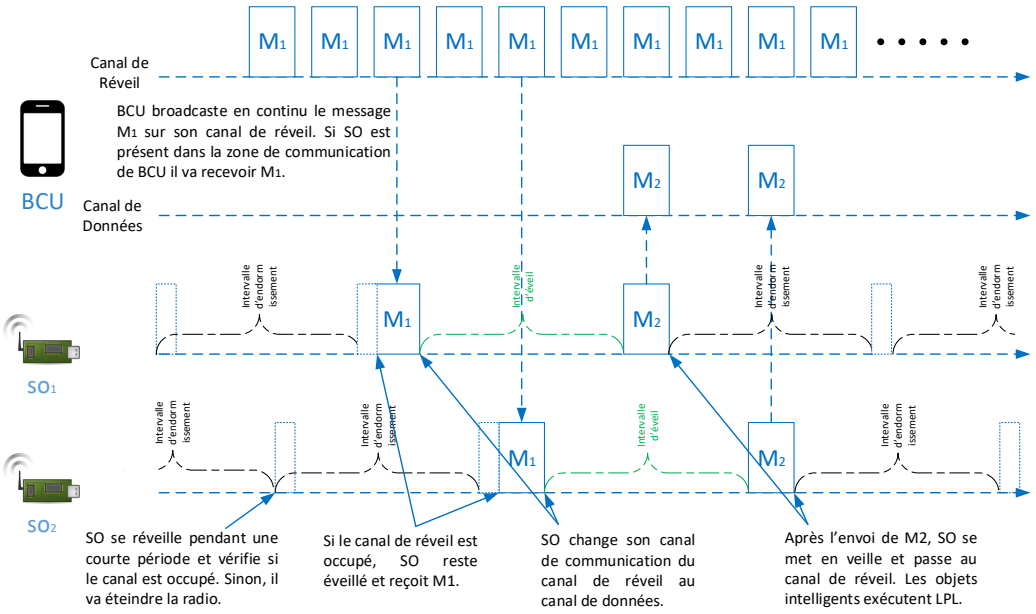


FIGURE 5.3 – Phase d'authentification et de collecte des données entre le BCU et les objets intelligents

Dans le scénario de collecte, le BCU diffuse en continu des signaux de réveil M_1 (contenant toutes les données vues précédemment, à savoir pk_{BCU} et $\hat{\sigma}_{BCU}$) sur le canal de réveil tout en se déplaçant (quand la personne bouge si le BCU est un smartphone) ; la fréquence à laquelle M_1 est envoyé doit être choisie de sorte à ce que les objets intelligents puissent le détecter dès leur réveil. En effet, le fonctionnement pour les objets intelligents est le suivant. Le SO est endormi la plupart du temps et il se réveille à intervalle régulier pour vérifier si le canal, appelé canal de réveil, est occupé. Si le canal est libre, le SO se rendort. Si en revanche le canal est occupé, cela signifie qu'une entité (ici le BCU) est à portée radio et essaye de communiquer avec lui. Le SO reste alors éveillé pour récupérer les données sur ce canal (c'est-à-dire le message M_1) et faire la suite des opérations, à savoir vérifier le contenu du message puis calculer et, après avoir reconfiguré sa radio sur le canal de données du BCU, envoyer ses données, c'est-à-dire M_2 (soit $pk_{SO}, C_{K_{SO}}(M), \hat{\sigma}_{SO}$).

Afin de ne pas surcharger la Figure 5.3, nous avons omis les acquittements du BCU auprès du SO sur le canal de données concernant le statut de la vérification de la validité de M_2 et du stockage en mémoire des données chiffrées reçues et des données associées (c'est-à-dire $C_{K_{SO}}(M)$ et S_{SO}). Une fois cet acquittement reçu, le SO pourra effacer les données correspondantes de sa mémoire et il pourra se rendormir.

5.4.8 Quelques mécanismes sécuritaires de notre proposition

Si sur la Figure 5.3, nous avons omis les transmissions accusant auprès du SO la bonne vérification (ou au contraire le rejet) par le BCU de σ_{SO}, $\hat{\sigma}_{SO}$, T_{SO} et le stockage en mémoire des données chiffrées et des données associées (c'est-à-dire $C_{K_{SO}}(M)$ et S_{SO}), il est nécessaire que ce message

soit présent. Dans notre solution, il est présent et un SO n'effacera pas les données M qu'il a envoyées sans cette confirmation d'écriture de la part du BCU. L'objet de ce message important est d'éviter une attaque qui existe dans l'article de Won et al. [205] où un attaquant peut intercepter le message M_1 d'un drone pour l'utiliser durant la période de validité afin de vider les mémoires des SO pour créer des dénis de services (l'attaquant ne pourra avoir accès aux données chiffrées envoyées par les SO mais il vide leur mémoire).

Mais même sans aller jusqu'à l'interception de M_1, dans le protocole de Won et al. [205], l'attaquant peut faire un déni de service sur les SO en leur faisant réaliser les étapes correspondant aux algorithmes SymmetricKeyGen et Encapsulation (cette dernière n'est pas présente dans notre protocole car intégré dans SymmetricKeyGen). En effet, en envoyant simplement un message M_1 fabriqué (avec des valeurs aléatoires, sauf pour le temps qui doit être cohérent), comme il n'y a pas d'authentification explicite que le message provient bien d'un BCU, dans leur protocole, le SO réalisera plusieurs multiplications scalaires, opérations très coûteuses en énergie, qui si elles sont répétées conduiront à vider sa batterie.

Dans l'article de Won et al. [205], si le drone est capturé alors l'attaquant peut lire toutes les données en mémoire. Dans notre proposition, les données restent chiffrées dans le BCU et même si un attaquant accède physiquement à ce dernier, il ne pourra pas les déchiffrer (sauf à casser les problèmes mathématiques sur lesquels repose la sécurité d'ECC). Seul le CGC pourra les déchiffrer, ce qui assure un niveau de confidentialité bien supérieure et c'est un point très important quand les données manipulées sont des données de santé.

Au lieu d'utiliser la technique de chiffrement du masque jetable, il est possible d'utiliser un algorithme de chiffrement par bloc, comme AES, et dans ce cas d'utiliser S_{SO} pour chiffrer plusieurs messages et non un seul comme dans la proposition présentée.

5.4.9 L'analyse du coût de calcul

Naturellement nous allons comparer le coût de notre protocole avec celui de Won et al. [205] et essentiellement en nombre de multiplications scalaires, opération la plus consommatrice en temps d'exécution et la plus gourmande en énergie.

Nous noterons SM l'opération de multiplication scalaire (en cohérence avec les notations du chapitre 3).

Dans notre protocole comme dans celui de Won et al. [205], les opérations à prendre en compte sont celles à partir de SymmetricKeyGen car les algorithmes précédents sont relatifs à la mise en place des clés du systèmes.

Dans notre protocole les opérations de vérification de σ_A et $\hat{\sigma}_A$ nécessitent $2SM$ chacune puisque que :

- pour vérifier qu'un $\sigma_A = (R_A, d_A)$ reçu est valide, il faut faire $d_A \cdot P \stackrel{?}{=} R_A + H(ID_{CGC}, ID_A, R_A, P_A, t_A) \cdot P_{CGC}$ (ID_{CGC} et P_{CGC} étant publiques)

- pour vérifier qu'un $\hat{\sigma}_A = (S_A, e_A, T_A)$ reçu est valide, il faut faire $e_A \cdot P \stackrel{?}{=} S_A + H(pk_A, T_A) \cdot P_A$ où pk_A reçu contient P_A.

Sur le BCU, mais aussi sur les SO, le calcul de $S_A = s_A \cdot P$ nécessite $1SM$ et sur le SO seulement, le calcul de la clé $K_{SO} = s_{SO} \cdot P_{CGC}$ nécessite encore $1SM$.

Au total, pour un BCU le nombre de multiplications scalaires est de 5 et pour le SO de 6. Dans le protocole de Won et al. [205], le BCU fait $6SM$ (2 dans les calculs de Y, de T et 4 dans la vérification de W) et le SO n'en fait que 4 (1 dans chaque calcul de U, de V, de Y et de T).

Ainsi si l'on ne regarde que ces chiffres notre solution pourrait sembler défavorable mais il faut prendre en compte toutes les attaques que nous avons contrées et les vérifications que nous avons mises en œuvre dans le protocole et qui permettent d'éviter de faire des calculs inutiles si un attaquant nous envoie de faux messages M_1. De plus, il est possible de ne faire les calculs de vérification des σ_A reçus qu'une fois et de réutiliser les résultats tant que la date d'expiration t_A n'est pas dépassée. Ceci permet de gagner $2SM$ tant du coté BCU que du coté SO, ce qui fait passer respectivement à $3SM$ et $4SM$. Toutefois, cette même technique peut s'appliquer aussi pour Won et al. [205] avec les mêmes durées de validité, en économisant $1SM$ sur le calcul de Y pour le SO et en économisant $2SM$ pour le BCU (1 dans le calcul de Y et 1 dans la vérification de W puisque $H_0(ID_A, R_A, P_A, t_A) \cdot P_{pub}$ est fixe tant que t_A n'a pas expiré), ce qui donne pour le BCU et le SO respectivement $4SM$ et $3SM$.

À nouveau, le léger surcoût sur le SO entre notre protocole et celui de référence doit être nuancé par le fait que nous controns de nombreuses attaques.

Le Tableau 5.1 résume les coûts de calcul des différents protocoles en version classique et optimisée en regard de leur résistance aux attaques identifiées sur le protocole de Won et al. [205].

TABLEAU 5.1 – Récapitulatif des coûts de calcul et de la résistance aux attaques pour différents protocoles

	Coût de calcul		**Résistance contre les attaques**			
Protocole	**SO (# SM)**	**BCU (# SM)**	**Vidage mémoire du SO**	**DoS sur les SO**	**Impersonation du BCU**	**Capture physique du BCU**
Won et al. [205]	4	6	✗	✗	✗	✗
Notre proposition	6	5	✓	✓	✓	✓
Won et al. [205] optimisé	3	4	✗	✗	✗	✗
Notre proposition optimisée	4	3	✓	✓	✓	✓

✓ : Supporté, ✗ : Non supporté

5.5 Conclusion

La proposition de protocole de communication sécurisée présentée dans ce chapitre permet donc à un BCU de faire une collecte sécurisée des données présentes sur les différents capteurs du WBAN et plus largement d'une maison intelligente afin de pouvoir les transmettre ultérieurement dans le cloud à une entité médicale autorisée (dans notre protocole le CGC). Si cette proposition a au premier abord un très léger surcoût en terme de calcul pour les objets intelligents, elle prend en compte les différentes attaques qui ne sont souvent pas considérées et elle reste très raisonnable en terme calculatoire grâce aux outils cryptographiques utilisés (courbes elliptiques et fonctions de hachage essentiellement).

L'intérêt de ce protocole est qu'il peut être utilisé dans de nombreux autres domaines d'applications qui auraient des contraintes fortes en terme de ressources de calcul et de sécurité.

Conclusion et perspectives

Dans cette thèse, nous avons abordé les problèmes de sécurité et de confidentialité relatifs aux systèmes ubiquitaires avec des contraintes de ressources strictes. Nous nous sommes principalement concentrés sur les protocoles d'authentification, qui sont des mécanismes nécessaires pour garantir que les entités communicantes sur le réseau sont bien légitimes. Dans ce cadre, nous avons dégagé deux contributions principales qui constituent le cœur de ce manuscrit. Notre première contribution comprend trois nouveaux protocoles d'authentification mutuelle adaptés aux ressources limitées des dispositifs IoT les plus contraints. Notre deuxième contribution consiste en un protocole d'identification et d'authentification mutuelle RFID léger pour les réseaux RFID qui s'appuie sur des opérations cryptographiques légères basées sur les fonctions de hachage unidirectionnelles sans collision.

Pour commencer, nous avons établi dans le chapitre 1 un état de l'art sur l'internet des objets appliqué au domaine médical. Plus précisément, nous nous sommes intéressés à la problématique de l'authentification dans le cadre de réseaux ubiquitaires déployés au sein de réseaux domiciliaires pour l'aide à l'autonomie de personnes dépendantes. Cela nous a permis de bien comprendre les différentes technologies, les critères et les contraintes des dispositifs intelligents (technologies de communication, sources d'énergie, capacité de calcul, etc.) ainsi que d'identifier les deux principales technologies de communication et d'interaction de l'IoT auxquelles nous avons appliqué nos travaux, à savoir, la technologie radio-fréquence (RFID) et les réseaux de capteurs (WSN).

Nous avons ensuite réalisé dans le chapitre 2 un état de l'art sur la sécurité IoT avec un prisme centré sur la notion de respect de la vie privée en particulier dans le domaine de la santé. Ceci nous a permis de définir les principales exigences de sécurité à respecter. Nous avons ensuite déterminé les mécanismes de sécurité à utiliser pour l'élaboration des protocoles de sécurité adaptés aux dispositifs IoT les plus contraints en ressources. Nous nous sommes intéressés aux protocoles d'authentification existants dans la littérature que nous avons classifiés selon deux modes de communication : Machine-to-cloud (M2C) et machine-to-machine (M2M). La nature des attaques possibles nous a aussi aidé à bien cerner les problèmes et les exigences de sécurité sur

lesquels reposeront nos propres contributions.

Le chapitre 3 présente notre première contribution qui consiste en trois protocoles d'authentification, dans le cadre de l'assistance à la vie ambiante. Les deux premiers se basent sur une communication M2C entre l'objet intelligent et le serveur (Cloud) tandis que le troisième est de nature décentralisée ; c'est-à-dire que la communication se fait en directement en M2M entre deux objets intelligents situés au domicile sans communication avec un serveur externe. Les trois protocoles se déroulent en deux phases, une phase de configuration et une phase d'authentification à trois étapes dans le but de réaliser l'authentification mutuelle des entités légitimes. Cette dernière phase est réalisée à chaque fois qu'une authentification est requise. La sécurité de nos trois protocoles se base sur l'utilisation de courbes elliptiques pour la génération de clés. Nous avons évalué la pertinence de nos propositions par le biais d'une évaluation complète des performances des dispositifs les plus limités en ressources selon l'opération la plus coûteuse utilisée dans les protocoles à savoir la multiplication scalaire. Ainsi, pour évaluer la validité de nos propositions, nous avons effectué des mesures de temps de différentes opérations dont la multiplication scalaire et ce, pour différentes courbes, sur des dispositifs Multos IoT Trust Anchor et Universal JCard. De plus, nous avons fourni une analyse de sécurité informelle pour démontrer que notre proposition est sécurisée contre diverses attaques. Deux vérifications formelles avec les outils AVISPA et ProVerif ont également été menées et ont montré que nos trois protocoles sont sûrs et répondent bien aux exigences de sécurité que nous nous étions fixées

Le chapitre 4 détaille notre deuxième contribution qui a consisté en un autre protocole d'identification et d'authentification RFID pour le domaine médical. Après un état de l'art centré sur les protocoles d'authentification RFID basés sur les fonctions de hachages unidirectionnelles sans collision, nous avons défini une nouvelle proposition n'utilisant pas cette fois-ci de courbes elliptiques. Elle s'appuie aussi sur des identités de groupe pour réduire la complexité de recherche lors de l'identification d'une étiquette RFID. Comme dans le chapitre précédent, nous avons réalisé une analyse informelle de sécurité reposant sur le modèle d'attaquant que nous avions défini et également sur des vérifications formelles de sécurité à l'aide des outils AVISPA et ProVerif. Ceci a démontré que notre protocole est sûr et sécurisé contre les attaques actives et passives supportées par ces deux outils. Enfin, nous avons réalisé une étude de performance qui nous a permis de mettre en évidence que notre protocole répondait aux failles de sécurité mises en évidence dans l'état de l'art mais également qu'il présentait une complexité moindre par rapport à d'autres protocoles similaires.

Enfin, le chapitre 5 présente en conclusion de ce manuscrit une troisième contribution en cours de finalisation. Ce protocole de communication sécurisée permet la collecte de données dans un réseau WBAN évoluant au sein d'un domicile. Ce protocole fait suite à la découverte d'at-

taques sur d'autres travaux de l'état de l'art et se veut comme une correction et une amélioration de ceux-ci.

Ces différents travaux ont permis de mettre en lumière le besoin d'amélioration des différents protocoles d'authentification existants pour assurer la confidentialité des échanges. Les différentes contributions que nous avons proposées s'appuient principalement sur la faible puissance de calcul des différents équipements que l'on peut retrouver dans un réseau domiciliaire comme les capteurs (portés ou ambiants) ou encore les étiquettes RFID. Ces solutions nous semblent adaptées pour les terminaux dont nous disposons actuellement. Ainsi, une des premières perspectives est de déployer les solutions présentées au chapitre 4 et au chapitre 5 sur des équipements réels pour compléter nos évaluations de performance et intégrer notamment la problématique de la consommation d'énergie dans nos métriques. À plus long terme, la puissance de calcul des capteurs augmentant très régulièrement, des solutions plus complexes et plus sécurisées devront être proposées. Ainsi, des solutions d'authentification reposant sur des solutions cryptographiques à base d'attributs ou centrées sur l'identité peuvent rapidement devenir des pistes intéressantes. De même, des solutions utilisant de la cryptographie post-quantique peuvent être des alternatives intéressantes aux solutions basées sur les courbes elliptiques que nous avons proposées.

Bibliographie

[1] Gildas Avoine and Philippe Oechslin. A scalable and provably secure hash-based rfid protocol. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 110–114. IEEE, 2005. xvi, 102, 103, 105

[2] Farzana Rahman, Md Zakirul Alam Bhuiyan, and Sheikh Iqbal Ahamed. A privacy preserving framework for rfid based healthcare systems. *Future Generation Computer Systems*, 72 :339–352, 2017. xvi, 103, 106, 107, 111, 116

[3] Ping Wang and Zhiping Zhou. An improved rfid authentication protocol based on group anonymous model. *Wireless Personal Communications*, 103(4) :2811–2831, 2018. xvi, 103, 107, 109, 116

[4] Klaus Finkenzeller. *RFID handbook : fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons, 2010. 3

[5] Vedat Coskun, Busra Ozdenizci, and Kerem Ok. A survey on near field communication (nfc) technology. *Wireless personal communications*, 71(3) :2259–2294, 2013. 3

[6] Jennifer Bray and Charles F Sturman. *Bluetooth 1.1 : connect without cables*. pearson Education, 2001. 3

[7] Jin-Shyan Lee, Yu-Wei Su, Chung-Chou Shen, et al. A comparative study of wireless protocols : Bluetooth, uwb, zigbee, and wi-fi. *Industrial electronics society*, 5 :46–51, 2007. 3, 16

[8] Konstantin Mikhaylov, Juha Petaejaejaervi, and Tuomo Haenninen. Analysis of capacity and scalability of the lora low power wide area network technology. In *European Wireless 2016; 22th European Wireless Conference*, pages 1–6. VDE, 2016. 3

[9] Luigi Atzori, Antonio Iera, and Giacomo Morabito. Understanding the internet of things : definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56 :122–140, 2017. 3, 6

[10] J Sathish Kumar and Dhiren R Patel. A survey on internet of things : Security and privacy issues. *International Journal of Computer Applications*, 90(11), 2014. 3, 41

[11] Mohamed Ahmed Hail and Stefan Fischer. Iot for aal : An architecture via information-centric networking. In *Globecom Workshops (GC Wkshps), 2015 IEEE*, pages 1–6. IEEE, 2015. 4

[12] Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. Security, privacy and trust in internet of things : The road ahead. *Computer Networks*, 76 :146–164, 2015. 4, 39

[13] Rongxing Lu, Xu Li, Xiaohui Liang, Xuemin Shen, and Xiaodong Lin. Grs : The green, reliability, and security of emerging machine to machine communications. *IEEE communications magazine*, 49(4), 2011. 4, 28

[14] Yosra Ben Saied, Alexis Olivereau, and Maryline Laurent. A distributed approach for secure m2m communications. In *New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*, pages 1–7. IEEE, 2012. 4, 28

[15] Wei Ren, Linchen Yu, Liangli Ma, and Yi Ren. Rise : a reliable and secure scheme for wireless machine to machine communications. *Tsinghua Science and Technology*, 18(1) :100–117, 2013. 4, 28

[16] Syed Umar Amin, M Shamim Hossain, Ghulam Muhammad, Musaed Alhussein, and Md Abdur Rahman. Cognitive smart healthcare for pathology detection and monitoring. *IEEE Access*, 7 :10745–10753, 2019. 4

[17] M Shamim Hossain, Ghulam Muhammad, and Atif Alamri. Smart healthcare monitoring : A voice pathology detection paradigm for smart cities. *Multimedia Systems*, pages 1–11, 2017. 4

[18] SM Riazul Islam, Daehan Kwak, MD Humaun Kabir, Mahmud Hossain, and Kyung-Sup Kwak. The internet of things for health care : a comprehensive survey. *IEEE Access*, 3 :678–708, 2015. 4

[19] Shivayogi Hiremath, Geng Yang, and Kunal Mankodiya. Wearable internet of things : Concept, architectural components and promises for person-centered healthcare. In *2014 4th International Conference on Wireless Mobile Communication and Healthcare-Transforming Healthcare Through Innovations in Mobile and Wireless Technologies (MOBIHEALTH)*, pages 304–307. IEEE, 2014. 4

[20] Mari Carmen Domingo. An overview of the internet of things for people with disabilities. *Journal of Network and Computer Applications*, 35(2) :584–596, 2012. 4

[21] David Metcalf, Sharlin TJ Milliard, Melinda Gomez, and Michael Schwartz. Wearables and the internet of things for health : Wearable, interconnected devices promise more efficient and comprehensive health care. *IEEE pulse*, 7(5) :35–39, 2016. 5

[22] Oana Geman, Saeid Sanei, Hariton-Nicolae Costin, Konstantinos Eftaxias, Oldřich Vyšata, Aleš Procházka, and Lenka Lhotská. Challenges and trends in ambient assisted living and intelligent tools for disabled and elderly people. In *2015 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM)*, pages 1–5. IEEE, 2015. 5

[23] P Suresh, J Vijay Daniel, V Parthasarathy, and RH Aswathy. A state of the art review on the internet of things (iot) history, technology and fields of deployment. In *2014 International conference on science engineering and management research (ICSEMR)*, pages 1–8. IEEE, 2014. 6

[24] Aurel-Dorian Floarea and Valentin Sgârciu. Smart refrigerator : A next generation refrigerator connected to the iot. In *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–6. IEEE, 2016. 6

[25] Dave Evans. The internet of things : How the next evolution of the internet is changing everything. *CISCO white paper*, 1(2011) :1–11, 2011. 6

[26] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot) : A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7) :1645–1660, 2013. 6

[27] John Krumm. *Ubiquitous computing fundamentals*. Chapman and Hall/CRC, 2016. 6

[28] Biljana L Risteska Stojkoska and Kire V Trivodaliev. A review of internet of things for smart home : Challenges and solutions. *Journal of Cleaner Production*, 140 :1454–1464, 2017. 8, 12

[29] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms : Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6) :599–616, 2009. 9

[30] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing : state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1) :7–18, 2010. 9

[31] M Reza Rahimi, Jian Ren, Chi Harold Liu, Athanasios V Vasilakos, and Nalini Venkatasubramanian. Mobile cloud computing : A survey, state of art and future directions. *Mobile Networks and Applications*, 19(2) :133–143, 2014. 9

[32] Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011. 9

[33] Farrukh Shahzad. State-of-the-art survey on cloud computing security challenges, approaches and solutions. *Procedia Computer Science*, 37 :357–362, 2014. 9

[34] Ibrar Yaqoob, Ejaz Ahmed, Ibrahim Abaker Targio Hashem, Abdelmuttlib Ibrahim Abdalla Ahmed, Abdullah Gani, Muhammad Imran, and Mohsen Guizani. Internet of things architecture : Recent advances, taxonomy, requirements, and open challenges. *IEEE wireless communications*, 24(3) :10–16, 2017. 11

[35] Louis Moreau, Emmanuel Conchon, and Damien Sauveron. Craft : A continuous remote attestation framework for iot. *IEEE Access*, 9 :46430–46447, 2021. 11

[36] Klara Nahrstedt, Hongyang Li, Phuong Nguyen, Siting Chang, and Long Vu. Internet of mobile things : Mobility-driven challenges, designs and implementations. In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 25–36. IEEE, 2016. 11, 12

[37] Michele Zorzi, Alexander Gluhak, Sebastian Lange, and Alessandro Bassi. From today's intranet of things to a future internet of things : a wireless-and mobility-related view. *IEEE Wireless communications*, 17(6) :44–51, 2010. 12

[38] Marci Meingast, Tanya Roosta, and Shankar Sastry. Security and privacy issues with health care information technology. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5453–5458. IEEE, 2006. 12

[39] Qi Jing, Athanasios V Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. Security of the internet of things : perspectives and challenges. *Wireless Networks*, 20(8) :2481–2501, 2014. 12

[40] V Chunduru and N Subramanian. Effects of power lines on performance of home control system. In *2006 International Conference on Power Electronic, Drives and Energy Systems*, pages 1–6. IEEE, 2006. 13

[41] Jaap C Haartsen. The bluetooth radio system. *IEEE personal communications*, 7(1) :28–36, 2000. 13

[42] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy : An emerging low-power wireless technology. *Sensors*, 12(9) :11734–11753, 2012. 14

[43] Ángela Hernández-Solana, David Perez-Diaz-de Cerio, Antonio Valdovinos, and Jose Luis Valenzuela. Proposal and evaluation of ble discovery process based on new features of bluetooth 5.0. *Sensors*, 17(9) :1988, 2017. 14

[44] Patricio Bulić, Gašper Kojek, and Anton Biasizzo. Data transmission efficiency in bluetooth low energy versions. *Sensors*, 19(17) :3746, 2019. 14

[45] Shahin Farahani. *ZigBee wireless networks and transceivers*. Newnes, 2011. 14

[46] Juan Carlos Zuniga and Benoit Ponsard. Sigfox system description. *LPWAN@ IETF97, Nov. 14th*, 25, 2016. 14

[47] Benny Vejlgaard, Mads Lauridsen, Huan Nguyen, István Z Kovács, Preben Mogensen, and Mads Sorensen. Coverage and capacity analysis of sigfox, lora, gprs, and nb-iot. In *2017 IEEE 85th vehicular technology conference (VTC Spring)*, pages 1–5. IEEE, 2017. 15

[48] Ankur Dumka. Smart information technology for universal healthcare. In *Healthcare Data Analytics and Management*, pages 211–226. Elsevier, 2019. 15

[49] Shadi Al-Sarawi, Mohammed Anbar, Kamal Alieyan, and Mahmood Alzubaidi. Internet of things (iot) communication protocols. In *2017 8th International conference on information technology (ICIT)*, pages 685–690. IEEE, 2017. 15

[50] Aloÿs Augustin, Jiazi Yi, Thomas Clausen, and William Townsley. A study of lora : Long range & low power networks for the internet of things. *Sensors*, 16(9) :1466, 2016. 15

[51] WEI Yang, LIU Yucheng, ZHU Hongxu, Chung Kit WU, and Kim Fung TSANG. A resilient safety lpwan for ubiquitous medical applications. In *2018 IEEE Symposium on Product Compliance Engineering-Asia (ISPCE-CN)*, pages 1–4. IEEE, 2018. 15

[52] A Valach and D Macko. Exploration of the lora technology utilization possibilities in healthcare iot devices. In *2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 623–628. IEEE, 2018. 15

[53] Afef Mdhaffar, Tarak Chaari, Kaouthar Larbi, Mohamed Jmaiel, and Bernd Freisleben. Iot-based health monitoring via lorawan. In *IEEE EUROCON 2017-17th International Conference on Smart Technologies*, pages 519–524. IEEE, 2017. 15

[54] Muneer Bani Yassein, Wail Mardini, and Ashwaq Khalil. Smart homes automation using z-wave protocol. In *2016 International Conference on Engineering & MIS (ICEMIS)*, pages 1–6. IEEE, 2016. 15

[55] Youssouf Zatout. Using wireless technologies for healthcare monitoring at home : A survey. In *2012 IEEE 14th international conference on e-Health networking, applications and services (Healthcom)*, pages 383–386. IEEE, 2012. 16

[56] Anil Yadav, Nitin Rakesh, Sujata Pandey, and Rajat K Singh. Development and analysis of iot framework for healthcare application. In *Advances in Computer and Computational Sciences*, pages 149–158. Springer, 2018. 16

[57] Darshana Thomas, Ross McPherson, Greig Paul, and James Irvine. Optimizing power consumption of wi-fi for iot devices : An msp430 processor and an esp-03 chip provide a power-efficient solution. *IEEE Consumer Electronics Magazine*, 5(4) :92–100, 2016. 16

[58] Serbulent Tozlu, Murat Senel, Wei Mao, and Abtin Keshavarzian. Wi-fi enabled sensors for internet of things : A practical approach. *IEEE Communications Magazine*, 50(6) :134–143, 2012. 16

[59] Marco Bassoli, Valentina Bianchi, and Ilaria De Munari. A plug and play iot wi-fi smart home system for human monitoring. *Electronics*, 7(9) :200, 2018. 16

[60] Jorge E Higuera and Jose Polo. Ieee 1451 standard in 6lowpan sensor networks using a compact physical-layer transducer electronic datasheet. *IEEE Transactions on Instrumentation and Measurement*, 60(8) :2751–2758, 2011. 17

[61] Tim Winter, Pascal Thubert, Anders Brandt, Jonathan Hui, Richard Kelsey, Philip Levis, Kris Pister, Rene Struik, Jean-Philippe Vasseur, and Roger Alexander. Rpl : Ipv6 routing protocol for low-power and lossy networks. Technical report, 2012. 17

[62] Thomas Clausen, Ulrich Herberg, and Matthias Philipp. A critical evaluation of the ipv6 routing protocol for low power and lossy networks (rpl). In *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 365–372. IEEE, 2011. 17

[63] Emilio Ancillotti, Raffaele Bruno, and Marco Conti. The role of the rpl routing protocol for smart grid communications. *IEEE Communications Magazine*, 51(1) :75–83, 2013. 18

[64] Fatma Gara, Leila Ben Saad, Rahma Ben Ayed, and Bernard Tourancheau. Rpl protocol adapted for healthcare and medical applications. In *2015 International wireless communications and mobile computing conference (IWCMC)*, pages 690–695. IEEE, 2015. 18

[65] Tuan Nguyen Gia, Nanda Kumar Thanigaivelan, Amir-Mohammad Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. Customizing 6lowpan networks towards internet-of-things based ubiquitous healthcare systems. In *2014 NORCHIP*, pages 1–6. IEEE, 2014. 18

[66] Farid Touati, Adel Ben Mnaouer, Ochirkhand Erdene-Ochir, Waiser Mehmood, Ammad Hassan, and Brahim Gaabab. Feasibility and performance evaluation of a 6lowpan-enabled platform for ubiquitous healthcare monitoring. *Wireless Communications and Mobile Computing*, 16(10) :1271–1281, 2016. 18

[67] Zach Shelby and Carsten Bormann. *6LoWPAN : The wireless embedded Internet*, volume 43. John Wiley & Sons, 2011. 18

[68] Dave Locke. Mq telemetry transport (mqtt) v3. 1 protocol specification. *IBM developerWorks Technical Library*, page 15, 2010. 18

[69] Ying-Tsung Lee, Wei-Hsuan Hsiao, Chin-Meng Huang, and T Chou Seng-cho. An integrated cloud-based smart home management system with community hierarchy. *IEEE Transactions on Consumer Electronics*, 62(1) :1–9, 2016. 18

[70] Meena Singh, MA Rajan, VL Shivraj, and P Balamuralidhar. Secure mqtt for internet of things (iot). In *2015 Fifth International Conference on Communication Systems and Network Technologies*, pages 746–751. IEEE, 2015. 19

[71] Zach Shelby, Klaus Hartke, and Carsten Bormann. The constrained application protocol (coap). Technical report, 2014. 19

[72] Marica Amadeo, Claudia Campolo, Antonio Iera, and Antonella Molinaro. Information centric networking in iot scenarios : The case of a smart home. In *2015 IEEE international conference on communications (ICC)*, pages 648–653. IEEE, 2015. 19

[73] Dinesh Thangavel, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan. Performance evaluation of mqtt and coap via a common middleware. In *2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)*, pages 1–6. IEEE, 2014. 19

[74] S Sujin Issac Samuel. A review of connectivity challenges in iot-smart home. In *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, pages 1–4. IEEE, 2016. 19

[75] Sarah A Al-Qaseemi, Hajer A Almulhim, Maria F Almulhim, and Saqib Rasool Chaudhry. Iot architecture challenges and issues : Lack of standardization. In *2016 Future Technologies Conference (FTC)*, pages 731–738. IEEE, 2016. 19

[76] Syed A Ahson and Mohammad Ilyas. *RFID handbook : applications, technology, security, and privacy*. CRC press, 2008. 20

[77] Lingfei Mo and Hongjian Zhang. Rfid antenna near the surface of metal. In *2007 International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, pages 803–806. IEEE, 2007. 21

[78] Amina Cherif. *Sécurité des RFID actifs et applications*. PhD thesis, Université Mouloud Mammeri Tizi Ouzou et Université de Limoges, 2016. 21

[79] Cristina Turcu Turcu et al. Rfid-based solutions for smarter healthcare. *arXiv preprint arXiv :1705.09855*, 2017. 21, 97

[80] Blake W Podaima, Marcia Friesen, and Robert D McLeod. A review of emerging smart rfid in healthcare. *CMBES Proceedings*, 33(1), 2018. 21, 97

[81] Moutaz Haddara and Anna Staaby. Rfid applications and adoptions in healthcare : A review on patient safety. *Procedia Computer Science*, 138 :80–88, 2018. 21, 97

[82] Yuan Zhang, Limin Sun, Houbing Song, and Xiaojun Cao. Ubiquitous wsn for healthcare : Recent advances and future prospects. *IEEE Internet of Things Journal*, 1(4) :311–318, 2014. 22

[83] Moshaddique Al Ameen, Jingwei Liu, and Kyungsup Kwak. Security and privacy issues in wireless sensor networks for healthcare applications. *Journal of medical systems*, 36(1) :93–101, 2012. 22

[84] Nidhi Sharma and Ravindara Bhatt. Privacy preservation in wsn for healthcare application. *Procedia computer science*, 132 :1243–1252, 2018. 22

[85] Guang-Zhong Yang and Guangzhong Yang. *Body sensor networks*, volume 1. Springer, 2006. 22

[86] YIN Yuehong, Yan Zeng, Xing Chen, and Yuanjie Fan. The internet of things in healthcare : An overview. *Journal of Industrial Information Integration*, 1 :3–13, 2016. 24

[87] Phillip A Laplante and Nancy Laplante. The internet of things in healthcare : Potential applications and challenges. *It Professional*, 18(3) :2–4, 2016. 24, 41

[88] Lili Liu, Eleni Stroulia, Ioanis Nikolaidis, Antonio Miguel-Cruz, and Adriana Rios Rincon. Smart homes and home health monitoring technologies for older adults : A systematic review. *International journal of medical informatics*, 91 :44–59, 2016. 24

[89] Kwang-il Kim, Shreya S Gollamudi, and Steven Steinhubl. Digital technology to enable aging in place. *Experimental Gerontology*, 88 :25–31, 2017. 24

[90] Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. Ambient intelligence in assisted living : enable elderly people to handle future interfaces. In *International conference on universal access in human-computer interaction*, pages 103–112. Springer, 2007. 25

[91] Mukhtiar Memon, Stefan Rahr Wagner, Christian Fischer Pedersen, Femina Hassan Aysha Beevi, and Finn Overgaard Hansen. Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. *Sensors*, 14(3) :4312–4341, 2014. 25

[92] Hong Sun, Vincenzo De Florio, Ning Gui, and Chris Blondia. Promises and challenges of ambient assisted living systems. In *2009 Sixth International Conference on Information Technology : New Generations*, pages 1201–1207. Ieee, 2009. 25

[93] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3) :579–590, 2012. 26

[94] Vladimir Villarreal, Jesús Fontecha, Ramón Hervás, and José Bravo. Ambient intelligence : technological solutions for wellness and supporting to daily activities. In *Latin American & Caribbean Conference, Panama-2012*, 2012. 26

[95] Carsten Röcker, Martina Ziefle, and Andreas Holzinger. Social inclusion in ambient assisted living environments : Home automation and convenience services for elderly user. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2011. 26

[96] Alessandro Andreadis and Riccardo Zambon. Ambient assisted living and digital inclusion : Overview of projects, services and interfaces. In *International Conference on Universal Access in Human-Computer Interaction*, pages 187–200. Springer, 2018. 26

[97] Diego Muñoz, Francisco J Gutierrez, and Sergio F Ochoa. Introducing ambient assisted living technology at the home of the elderly : challenges and lessons learned. In *International Work-Conference on Ambient Assisted Living*, pages 125–136. Springer, 2015. 26

[98] Ana Isabel Martins, Alexandra Queirós, and Nelson Pacheco da Rocha. Ambient assisted living : Introduction and overview. In *Usability, Accessibility and Ambient Assisted Living*, pages 1–12. Springer, 2018. 26

[99] Stephanie Blackman, Claudine Matlo, Charisse Bobrovitskiy, Ashley Waldoch, Mei Lan Fang, Piper Jackson, Alex Mihailidis, Louise Nygård, Arlene Astell, and Andrew Sixsmith. Ambient assisted living technologies for aging well : a scoping review. *Journal of Intelligent Systems*, 25(1) :55–69, 2016. 27

[100] Alexandra Queiros, Ana Dias, Anabela G Silva, and Nelson Pacheco Rocha. Ambient assisted living and health-related outcomes—a systematic literature review. In *Informatics*, volume 4, page 19. Multidisciplinary Digital Publishing Institute, 2017. 27

[101] Eftychios G Christoforou, Andreas S Panayides, Sotiris Avgousti, Panicos Masouras, and Constantinos S Pattichis. An overview of assistive robotics and technologies for elderly care. In *Mediterranean Conference on Medical and Biological Engineering and Computing*, pages 971–976. Springer, 2019. 27

[102] Arsénio Reis, Hugo Paredes, Isabel Barroso, Maria João Monteiro, Vitor Rodrigues, Salik Ram Khanal, and João Barroso. Autonomous systems to support social activity of elderly people a prospective approach to a system design. In *2016 1st International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, pages 1–5. IEEE, 2016. 27

[103] Rubén Blasco, Álvaro Marco, Roberto Casas, Diego Cirujano, and Richard Picking. A smart kitchen for ambient assisted living. *Sensors*, 14(1) :1629–1653, 2014. 27

[104] Tommaso Magherini, Alessandro Fantechi, Chris D Nugent, and Enrico Vicario. Using temporal logic and model checking in automated recognition of human activities for ambient-assisted living. *IEEE Transactions on Human-Machine Systems*, 43(6) :509–521, 2013. 27

[105] Valentina Bianchi, Ferdinando Grossi, Ilaria De Munari, and Paolo Ciampolini. Multi sensor assistant : a multisensor wearable device for ambient assisted living. *Journal of Medical Imaging and Health Informatics*, 2(1) :70–75, 2012. 27

[106] Arie Reichman and Moti Zwiling. The architecture of ambient assisted living system. In *2011 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS 2011)*, pages 1–4. IEEE, 2011. 27

[107] Samir V Zanjal and Girish R Talmale. Medicine reminder and monitoring system for secure health using iot. *Procedia Computer Science*, 78 :471–476, 2016. 32

[108] Jorge Gómez, Byron Oviedo, and Emilio Zhuma. Patient monitoring system based on internet of things. *Procedia Computer Science*, 83 :90–97, 2016. 32

[109] C. Shepherd, G. Arfaoui, I. Gurulian, R. P. Lee, K. Markantonakis, R. N. Akram, D. Sauveron, and E. Conchon. Secure and trusted execution : Past, present, and future - a critical review in the context of the internet of things and cyber-physical systems. In *2016 IEEE Trustcom/-BigDataSE/ISPA*, pages 168–177, 2016. 33

[110] Ahmad Ibrahim, Ahmad-Reza Sadeghi, and Gene Tsudik. Us-aid : Unattended scalable attestation of iot devices. In *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*, pages 21–30. IEEE, 2018. 33

[111] N. Asokan, Ferdinand Brasser, Ahmad Ibrahim, Ahmad-Reza Sadeghi, Matthias Schunter, Gene Tsudik, and Christian Wachsmann. Seda : Scalable embedded device attestation. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 964–975, New York, NY, USA, 2015. ACM. 33

[112] Mahmoud Ammar, Bruno Crispo, and Gene Tsudik. Simple : A remote attestation approach for resource-constrained iot devices. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, pages 247–258. IEEE, 2020. 33

[113] YongBin Zhou and DengGuo Feng. Side-channel attacks : Ten years after its publication and the impacts on cryptographic module security testing. *IACR Cryptol. ePrint Arch.*, 2005 :388, 2005. 34

[114] A. Abu-Mahfouz and G. P. Hancke. Distance bounding : A practical security solution for real-time location systems. *IEEE Transactions on Industrial Informatics*, 9(1) :16–27, 2013. 36

[115] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2) :198–208, 1983. 36, 60, 157

[116] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao. A survey on security and privacy issues in internet-of-things. *IEEE Internet of Things Journal*, 4(5) :1250–1258, 2017. 39

[117] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things : Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5) :1125–1142, 2017. 39

[118] Ton van Deursen. 50 ways to break rfid privacy. In *IFIP PrimeLife International Summer School on Privacy and Identity Management for Life*, pages 192–205. Springer, 2010. 40

[119] Jan Henrik Ziegeldorf, Oscar Garcia Morchon, and Klaus Wehrle. Privacy in the internet of things : threats and challenges. *Security and Communication Networks*, 7(12) :2728–2742, 2014. 40

[120] Na Li, Nan Zhang, Sajal K Das, and Bhavani Thuraisingham. Privacy preservation in wireless sensor networks : A state-of-the-art survey. *Ad Hoc Networks*, 7(8) :1501–1514, 2009. 40

[121] Pratap Misra and Per Enge. Global positioning system : signals, measurements and performance second edition. *Massachusetts : Ganga-Jamuna Press*, 2006. 40

[122] Mireille Hildebrandt. Profiling : From data to knowledge. *Datenschutz und Datensicherheit-DuD*, 30(9) :548–552, 2006. 41

[123] Christoph P Mayer. Security and privacy challenges in the internet of things. *Electronic Communications of the EASST*, 17, 2009. 41

[124] Liane Margarida Rockenbach Tarouco, Leandro Márcio Bertholdo, Lisandro Zambenedetti Granville, Lucas Mendes Ribeiro Arbiza, Felipe Carbone, Marcelo Marotta, and Jose Jair Cardoso De Santanna. Internet of things in healthcare : Interoperatibility and security issues. In *2012 IEEE international conference on communications (ICC)*, pages 6121–6125. IEEE, 2012. 41

[125] Orlando Arias, Jacob Wurm, Khoa Hoang, and Yier Jin. Privacy and security in internet of things and wearable devices. *IEEE Transactions on Multi-Scale Computing Systems*, 1(2) :99–109, 2015. 41

[126] Laurie Pycroft and Tipu Z Aziz. Security of implantable medical devices with wireless connections : The dangers of cyber-attacks, 2018. 42

[127] Darlene Storm. Medjack : Hackers hijacking medical devices to create backdoors in hospital networks. *Computerworld*, 8, 2015. 42

[128] Eduard Marin, Dave Singelée, Flavio D Garcia, Tom Chothia, Rik Willems, and Bart Preneel. On the (in) security of the latest generation implantable cardiac defibrillators and how to secure them. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 226–236. ACM, 2016. 42

[129] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Annual international cryptology conference*, pages 1–15. Springer, 1996. 44

[130] Robert R Jueneman, Stephen M Matyas, and Carl H Meyer. Message authentication with manipulation detection code. In *1983 IEEE Symposium on Security and Privacy*, pages 33–33. IEEE, 1983. 44

[131] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 33–43. ACM, 1989. 44

[132] Ralph C Merkle. One way hash functions and des. In *Conference on the Theory and Application of Cryptology*, pages 428–446. Springer, 1989. 44

[133] Gustavus J Simmons. Symmetric and asymmetric encryption. *ACM Computing Surveys (CSUR)*, 11(4) :305–330, 1979. 45

[134] Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O'neill. Order-preserving symmetric encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 224–241. Springer, 2009. 45

[135] Vincent Rijmen and Joan Daemen. Advanced encryption standard. *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, pages 19–22, 2001. 46

[136] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Annual International Cryptology Conference*, pages 537–554. Springer, 1999. 46

[137] Monika Agrawal and Pradeep Mishra. A comparative survey on symmetric key encryption techniques. *International Journal on Computer Science and Engineering*, 4(5) :877, 2012. 46

[138] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2) :120–126, 1978. 46

[139] Gurpreet Singh. A study of encryption algorithms (rsa, des, 3des and aes) for information security. *International Journal of Computer Applications*, 67(19), 2013. 47

[140] Xin Zhou and Xiaofei Tang. Research and implementation of rsa algorithm for encryption and decryption. In *Proceedings of 2011 6th International Forum on Strategic Technology*, volume 2, pages 1118–1121. IEEE, 2011. 47

[141] Darrel Hankerson and Alfred Menezes. *Elliptic curve cryptography*. Springer, 2011. 48

[142] Neal Koblitz, Alfred Menezes, and Scott Vanstone. The state of elliptic curve cryptography. *Designs, codes and cryptography*, 19(2-3) :173–193, 2000. 48

[143] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling Chang Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In *International workshop on cryptographic hardware and embedded systems*, pages 119–132. Springer, 2004. 49

[144] Kristin Lauter. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless communications*, 11(1) :62–67, 2004. 49

[145] H Ong, Claus-Peter Schnorr, and Adi Shamir. An efficient signature scheme based on quadratic equations. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 208–216. ACM, 1984. 50, 61, 75

[146] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252. Springer, 1989. 50

[147] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2) :77–94, 1988. 50

[148] Amos Fiat and Adi Shamir. How to prove yourself : Practical solutions to identification and signature problems. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 186–194. Springer, 1986. 50

[149] Hartwig Mayer. Ecdsa security in bitcoin and ethereum : a research survey. *CoinFaabrik, June*, 28 :126, 2016. 50

[150] Pawani Porambage, An Braeken, Andrei Gurtov, Mika Ylianttila, and Susanna Spinsante. Secure end-to-end communication for constrained devices in iot-enabled ambient assisted living systems. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 711–714. IEEE, 2015. 52, 54

[151] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11) :612–613, 1979. 52

[152] Kim Thuat Nguyen, Nouha Oualha, and Maryline Laurent. Authenticated key agreement mediated by a proxy re-encryptor for the internet of things. In *European Symposium on Research in Computer Security*, pages 339–358. Springer, 2016. 52

[153] David Nuñez, Isaac Agudo, and Javier Lopez. Attacks to a proxy-mediated key agreement protocol based on symmetric encryption. *IACR Cryptology ePrint Archive*, 2016 :1081, 2016. 52

[154] Chunhua Jin, Chunxiang Xu, Xiaojun Zhang, and Fagen Li. A secure ecc-based rfid mutual authentication protocol to enhance patient medication safety. *Journal of medical systems*, 40(1) :12, 2016. 52, 60, 62, 70, 80, 81, 82

[155] Zhenguo Zhao. A secure rfid authentication protocol for healthcare environments using elliptic curve cryptosystem. *Journal of medical systems*, 38(5) :46, 2014. 53, 80, 81, 82

[156] Amjad Ali Alamr, Firdous Kausar, Jongsung Kim, and Changho Seo. A secure ecc-based rfid mutual authentication protocol for internet of things. *The Journal of Supercomputing*, pages 1–14, 2016. 53, 80, 81, 82

[157] Yi-Pin Liao and Chih-Ming Hsiao. A secure ecc-based rfid authentication scheme using hybrid protocols. In *Advances in Intelligent Systems and Applications-Volume 2*, pages 1–13. Springer, 2013. 53, 80, 81, 82

[158] Negin Dinarvand and Hamid Barati. An efficient and secure rfid authentication protocol using elliptic curve cryptography. *Wireless Networks*, pages 1–14, 2017. 53, 80, 81, 82

[159] Dhruti Sharma and Devesh Jinwala. Functional encryption in iot e-health care system. In *International Conference on Information Systems Security*, pages 345–363. Springer, 2015. 53

[160] Ruixuan Li, Chenglin Shen, Heng He, Zhiyong Xu, and Cheng-Zhong Xu. A lightweight secure data sharing scheme for mobile cloud computing. *IEEE Transactions on Cloud Computing*, 2017. 53

[161] Pawani Porambage, An Braeken, Pardeep Kumar, Andrei Gurtov, and Mika Ylianttila. Proxy-based end-to-end key establishment protocol for the internet of things. In *Communication Workshop (ICCW), 2015 IEEE International Conference on*, pages 2677–2682. IEEE, 2015. 54

[162] Ruhul Amin, SK Hafizul Islam, GP Biswas, Muhammad Khurram Khan, and Neeraj Kumar. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Generation Computer Systems*, 80 :483–495, 2018. 54

[163] Qi Jiang, Jianfeng Ma, Chao Yang, Xindi Ma, Jian Shen, and Shehzad Ashraf Chaudhry. Efficient end-to-end authentication protocol for wearable health monitoring systems. *Computers & Electrical Engineering*, 63 :182–195, 2017. 54

[164] Gurkan Tuna, Dimitrios G. Kogias, V. Cagri Gungor, Cengiz Gezer, Erhan Taşkın, and Erman Ayday. A survey on information security threats and solutions for machine to machine (m2m) communications. *Journal of Parallel and Distributed Computing*, 109 :142 – 154, 2017. 54

[165] M. Saadeh, A. Sleit, M. Qatawneh, and W. Almobaideen. Authentication techniques for the internet of things : A survey. In *2016 Cybersecurity and Cyberforensics Conference (CCC)*, pages 28–34, Aug 2016. 54

[166] Mohamed Amine Ferrag, Leandros A Maglaras, Helge Janicke, Jianmin Jiang, and Lei Shu. Authentication protocols for internet of things : a comprehensive survey. *Security and Communication Networks*, 2017, 2017. 54

[167] Tuomas Aura and Pekka Nikander. Stateless connections. In *International Conference on Information and Communications Security*, pages 87–97. Springer, 1997. 59

[168] TA Team et al. Avispa v1. 1 user manual. *Information Society Technologies Programme (June 2006), http ://avispa-project. org*, 2006. 62, 112

[169] Bruno Blanchet. Automatic verification of security protocols in the symbolic model : The verifier proverif. In *Foundations of Security Analysis and Design VII*, pages 54–87. Springer, 2014. 62, 112

[170] Pierre-François Bonnefoi, Pierre Dusart, Damien Sauveron, Raja Naeem Akram, and Konstantinos Markantonakis. A set of efficient privacy protection enforcing lightweight authentication protocols for low-cost rfid tags. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 612–620. IEEE, 2015. 64

[171] Y. Zhuang, A. Yang, G. P. Hancke, D. S. Wong, and G. Yang. Energy-efficient distance-bounding with residual charge computation. *IEEE Transactions on Emerging Topics in Computing*, pages 1–1, 2018. 67

[172] MULTOS Limited. Multos Trust Anchor. https://www.multos.com/trust_anchor. Online; accessed 20 May 2019. 83

[173] Universal Smart Cards Limited. Universal JCard. http://www.universaljcard.com/. Online; accessed 20 May 2019. 84

[174] MULTOS Limited. Multos Consortium. https://www.multos.com/consortium. Online; accessed 20 May 2019. 84

[175] Infineon Technologies AG. SLE 78CLUFX5000PH. https://www.infineon.com/cms/en/product/security-smart-card-solutions/security-controllers-for-usb-tokens/sle-78clufx5000ph/. Online; accessed 20 May 2019. 84

[176] Infineon Technologies AG. SLE 78CLFX4000P. https://www.infineon.com/cms/en/product/security-smart-card-solutions/security-controllers/sle-78/sle-78clfx4000p/. Online; accessed 20 May 2019. 84

[177] Patrick Van haver. Unveiling Java Card 3.1 : New I/O and Trusted Peripherals. https://blogs.oracle.com/javaiot/unveiling-java-card-31%3a-new-io-and-trusted-peripherals, November 2018. Online; accessed 20 May 2019. 84

[178] Oracle Inc. Oracle Java Card Boosts Security for IoT Devices at the Edge. https://www.oracle.com/corporate/pressrelease/oracle-java-card-boosts-security-011619.html, January 2019. Online; accessed 20 May 2019. 84

[179] Oracle Inc. Java Card 3.1 Documentation. https://docs.oracle.com/en/java/javacard/3.1/, January 2019. Online; accessed 20 May 2019. 84

[180] Infineon Technologies AG. Integrity Guard. https://www.infineon.com/dgdl/Infineon-Integrity_Guard_The_smartest_digital_security_technology_in_the_industry_06.18-WP-v01_01-EN.pdf?fileId=5546d46255dd933d0155e31c46fa03fb, June 2018. Online; accessed 20 May 2019. 84

[181] Infineon Technologies AG. SOLID FLASH Chip Card Controllers. https://www.infineon.com/dgdl/Infineon-SOLID_FLASH_Chip_Card_Controllers-PB-v04_12-EN.pdf?fileId=5546d46255dd933d0155e31d0ac105d9, May 2012. Online; accessed 20 May 2019. 84

[182] MULTOS Limited. Multos IoT Trust Anchor Developer Boards. https://www.multos.com/dev_boards. Online; accessed 20 May 2019. 85

[183] MULTOS Limited. ML5-P22 AND MC5-P22 ON INFINEON SLE78 PLATFORM. https://www.multos.com/products/approved_platforms/MIR/multos_international/m5-p22. Online; accessed 20 May 2019. 85

[184] MULTOS Limited. ML5-P19 AND MC5-P19 ON INFINEON SLE78 PLATFORM . https://www.multos.com/products/approved_platforms/MIR/multos_international/m5_p19. Online; accessed 20 May 2019. 85

[185] MULTOS Limited. SmartDeck. https://www.multos.com/software. Online; accessed 20 May 2019. 85

[186] Oracle Inc. Java Card Technology. https://www.oracle.com/technetwork/java/embedded/javacard/overview/index.html. Online; accessed 20 May 2019. 85

[187] Damien Sauveron. Multiapplication smart card : Towards an open smart card? *Information Security Technical Report*, 14(2) :70 – 78, 2009. Smart Card Applications and Security. 85

[188] Infineon Technologies AG. Java Card IDE. https://www.infineon.com/cms/en/product/promopages/devkit4ID/. Online; accessed 20 May 2019. 85

[189] Oracle Inc. Java Card Development Kit. https://www.oracle.com/technetwork/java/embedded/javacard/downloads/javacard-sdk-2043229.html. Online; accessed 20 May 2019. 85

[190] Fatma Merabet, Amina Cherif, Malika Belkadi, Olivier Blazy, Emmanuel Conchon, and Damien Sauveron. New efficient M2C and M2M mutual authentication protocols for iot-based healthcare applications. *Peer Peer Netw. Appl.*, 13(2) :439–474, 2020. 88

[191] Miyako Ohkubo, Koutarou Suzuki, Shingo Kinoshita, et al. Cryptographic approach to "privacy-friendly" tags. In *RFID privacy workshop*, volume 82. Cambridge, USA, 2003. 102, 104

[192] Gene Tsudik. Ya-trap : Yet another trivial rfid authentication protocol. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 4–pp. IEEE, 2006. 102

[193] Ari Juels and Stephen A Weis. Defining strong privacy for rfid. *ACM Transactions on Information and System Security (TISSEC)*, 13(1) :7, 2009. 102

[194] Keerti Srivastava, Amit K Awasthi, Sonam D Kaul, and RC Mittal. A hash based mutual rfid tag authentication protocol in telecare medicine information system. *Journal of medical systems*, 39(1) :153, 2015. 102

[195] Chun-Ta Li, Chi-Yao Weng, and Cheng-Chi Lee. A secure rfid tag authentication protocol with privacy preserving in telecare medicine information system. *Journal of medical systems*, 39(8) :77, 2015. 102

[196] Mustapha Benssalah, Mustapha Djeddou, and Karim Drouiche. Dual cooperative rfid-telecare medicine information system authentication protocol for healthcare environments. *Security and Communication Networks*, 9(18) :4924–4948, 2016. 102

[197] Neeraj Kumar, Kuljeet Kaur, Subhas C Misra, and Rahat Iqbal. An intelligent rfid-enabled authentication scheme for healthcare applications in vehicular mobile cloud. *Peer-to-Peer Networking and Applications*, 9(5) :824–840, 2016. 102

[198] Fan Wu, Lili Xu, Saru Kumari, Xiong Li, Ashok Kumar Das, and Jian Shen. A lightweight and anonymous rfid tag authentication protocol with cloud assistance for e-healthcare applications. *Journal of Ambient Intelligence and Humanized Computing*, 9(4) :919–930, 2018. 103

[199] Seyed Mohammad Alavi, Karim Baghery, Behzad Abdolmaleki, and Mohammad Reza Aref. Traceability analysis of recent rfid authentication protocols. *Wireless Personal Communications*, 83(3) :1663–1682, 2015. 103, 107

[200] Kai Fan, Wei Jiang, Hui Li, and Yintang Yang. Lightweight rfid protocol for medical privacy protection in iot. *IEEE Transactions on Industrial Informatics*, 14(4) :1656–1665, 2018. 103

[201] Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In *Annual International Cryptology Conference*, pages 617–630. Springer, 2003. 104

[202] Martin Hellman. A cryptanalytic time-memory trade-off. *IEEE transactions on Information Theory*, 26(4) :401–406, 1980. 104

[203] M Udin Harun Al Rasyid, Bih-Hwang Lee, and Amang Sudarsono. Wireless body area network for monitoring body temperature, heart beat and oxygen in blood. In *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pages 95–98. IEEE, 2015. 122

[204] Amna Asif and Irshad Ahmed Sumra. Applications of wireless body area network (wban) : A survey. *Engineering Science and Technoogy. International. Research Journal*, pages 64–71, 2017. 122

[205] Jongho Won, Seung-Hyun Seo, and Elisa Bertino. A secure communication protocol for drones and smart objects. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 249–260, 2015. 122, 123, 125, 129, 130

[206] Prabha Selvaraj and Sumathi Doraikannan. Privacy and security issues on wireless body area and iot for remote healthcare monitoring. *Intelligent Pervasive Computing Systems for Smarter Healthcare*, pages 227–253, 2019. 123

[207] Min Chen, Sergio Gonzalez, Athanasios Vasilakos, Huasong Cao, and Victor CM Leung. Body area networks : A survey. *Mobile networks and applications*, 16(2) :171–193, 2011. 123

[208] Samaher Al-Janabi, Ibrahim Al-Shourbaji, Mohammad Shojafar, and Shahaboddin Shamshirband. Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications. *Egyptian Informatics Journal*, 18(2) :113–122, 2017. 123

[209] Lauren Linkous, Nasibeh Zohrabi, and Sherif Abdelwahed. Health monitoring in smart homes utilizing internet of things. In *2019 IEEE/ACM International Conference on Connected Health : Applications, Systems and Engineering Technologies (CHASE)*, pages 29–34. IEEE, 2019. 123

[210] Zhaoyang Zhang, Honggang Wang, Athanasios V Vasilakos, and Hua Fang. Ecg-cryptography and authentication in body area networks. *IEEE Transactions on Information Technology in Biomedicine*, 16(6) :1070–1078, 2012. 123

[211] Xiong Li, Maged Hamada Ibrahim, Saru Kumari, Arun Kumar Sangaiah, Vidushi Gupta, and Kim-Kwang Raymond Choo. Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. *Computer Networks*, 129 :429–443, 2017. 123

[212] Ashok Kumar Das, Mohammad Wazid, Neeraj Kumar, Muhammad Khurram Khan, Kim-Kwang Raymond Choo, and YoungHo Park. Design of secure and lightweight authentication protocol for wearable devices environment. *IEEE journal of biomedical and health informatics*, 22(4) :1310–1322, 2017. 123

[213] Ankur Gupta, Meenakshi Tripathi, Tabish Jamil Shaikh, and Aakar Sharma. A lightweight anonymous user authentication and key establishment scheme for wearable devices. *Computer Networks*, 149 :29–42, 2019. 123

[214] MyeongHyun Kim, JoonYoung Lee, SungJin Yu, KiSung Park, YoHan Park, and YoungHo Park. A secure authentication and key establishment scheme for wearable devices. In *2019 28th*

International Conference on Computer Communication and Networks (ICCCN), pages 1–2. IEEE, 2019. 123

[215] David Moss, Jonathan Hui, and Kevin Klues. Low power listening. *TinyOS Core Working Group, TEP*, 105 :29, 2007. 127

Annexe A

Vérification formelle de protocoles en utilisant l'outil AVISPA

A.1 AVISPA

Cette annexe présente la validation formelle de quatre protocoles en utilisant l'outil « Automated Validation of Internet Protocols and Applications » (AVISPA). AVISPA fournit un langage formel modulaire et expressif permettant de spécifier les protocoles ainsi que leurs propriétés de sécurité. Il intègre différents systèmes dits « back-end » qui permettent d'implémenter diverses techniques d'analyse automatique.

AVISPA utilise un langage de spécification de protocole de haut niveau nommé « High-Level Protocol Specification Language » (HLPSL) qui prend en charge la description de protocoles, des objectifs de sécurité et des hypothèses concernant l'environnement. Il se caractérise par une modularité accrue qui lui permet de spécifier des modèles complexes de flux de contrôle, des structures de données et différents modèles d'intrus. HLPSL permet d'écrire les spécifications du protocole de sécurité en utilisant différents rôles. Certains rôles, appelés rôles de base, servent à décrire les actions d'un seul agent dans l'exécution d'un protocole ou d'un sous-protocole. D'autres, appelés rôles composés, instancient les rôles de base pour modéliser une exécution complète du protocole, une session du protocole entre plusieurs agents ou le modèle de protocole.

La version courante d'AVISPA intègre quatre back-ends : On-the-Fly Model-Checker (OFMC), CL-based Model-Checker (CL-AtSe), SAT-based Model-Checker (SATMC) et Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). Le modèle de l'attaquant considéré par AVISPA est le modèle de Dolev-Yao [115] c'est-à-dire que l'attaquant a accès au canal de communication et il peut écouter ou intercepter les communications, injecter de nou-

veaux messages ou modifier les messages en transit.

Les protocoles que nous validons dans cette annexe sont l'amélioration du protocole d'authentification M2C de Jin et al. (en section A.2), le protocole d'authentification M2C basé sur les fonctions de hachage (en section A.3) et le protocole d'authentification M2M basé sur ECC (en section A.4), discutés dans le chapitre 3 ainsi que le protocole d'authentification RFID basé sur le hachage (en section A.5) discuté dans le chapitre 4.

A.2 Amélioration du protocole d'authentification M2C de Jin et al.

Il existe deux rôles de base S et T qui expliquent l'activité de *Server* et T_i.

- Il existe deux agents `T` et `S` qui utilisent une fonction de hachage `H`, une fonction de multiplication modulaire `MULT`, une fonction de multiplication scalaire `MECC`, une fonction d'addition modulaire `ADD`, une fonction d'addition de points `ADDP`, un générateur de courbe `P` plus l'identité d'un objet intelligent `T Idt`, le résultat du hachage de son identité `HidT` et la clé publique du serveur `Ps` pour calculer les messages d'authentification `AUTHI` et `AUTHS`.
- La confidentialité (le secret, « secrecy ») de l'identité de l'objet intelligent T `Idt`, du résultat son hachage `HidT`, de la clé secrète du serveur `Xs` et des nombres aléatoires secrets `Ri`, `Rs` est modélisée à l'aide des prédicats `secret(Idt, sec_idt, T)`, `secret(Idt, sec_idt, S)`, `secret(HidT, sec_hidt, T)`, `secret(HidT, sec_hidt, S)`, `secret(Xs, sec_xs, S)`, `secret(Ri', sec_ri, T)`, `secret(Rs', sec_rs, S)` qui sont gérés par le `protocol_id` : respectivement `sec_idt`, `sec_hidt`, `sec_xs`, `sec_ri` et `sec_rs`. Les paramètres `Idt`, `HidT`, `Xs` et `Ri`, `Rs` sont gardés secrets pour `T` et `S`.
- L'authentification mutuelle est réalisée via les objectifs `witness` et `request`, c'est-à-dire `witness(T, S, auth_i, AUTHI')`, `request(S, T, auth_i, AUTHI')`, `witness(S, T, auth_s, AUTHS')`, `request (T, S, auth_s, AUTHS')`. `witness(T, S, auth_i, AUTHI')` déclare que l'agent `T` prétend être l'homologue de l'agent `S`, en accord sur la valeur `AUTHI'`. `auth_i` représente l'authentification de `AUTHI'` indiquée dans la section objectif, alors que `request(S, T, auth_i, AUTHI')` déclare que l'agent `S` accepte la valeur `AUTHI'` et s'appuie désormais sur la garantie que l'agent `T` existe et est d'accord avec lui à propos de cette valeur.
- L'environnement contient les constantes globales et la composition d'une ou plusieurs sessions. L'intrus participe en tant que session concrète à l'exécution du protocole.

Listing A.1 – Spécification AVISPA

```
role serverS (S, T:agent,
              P, Xs, Idt, HidT: text,
              Ps: message,
              MECC, MULT, H, ADD, ADDP: hash_func,
              SND, RCV : channel(dy))
played_by   S
def=
local State: nat,
          Rs: text,
          RI, RS, MASK, AUTHS, AUTHI, E, M: message
const sec_rs, sec_xs, sec_idt, sec_hidt, auth_s, auth_i: protocol_id
```

```
init State:=0
transition
1. State=0 /\ RCV(start) =|> State':=2 /\ Rs':=new()
                                       /\ RS':=MECC(Rs', P)
                                       /\ secret(Rs', sec_rs, S)
                                       /\ SND(RS')
2. State=2 /\ RCV(RI', AUTHI', M')
           /\ MASK'=MECC(Xs, RI')
           /\ HidT'=xor(M', H(MASK'))
           /\ HidT=H(Idt')
           /\ Idt =xor(AUTHI', H(RS, MASK'))
           /\ Idt'=Idt =|> State':=4 /\ E':=H(RS, RI, Idt)
                                    /\ AUTHS':=MECC(ADD(MULT(Xs, E'), Rs), P)
                                    /\ SND(AUTHS')
                                    /\ secret(Xs, sec_xs, S)
                                    /\ secret(Idt, sec_idt, S)
                                    /\ secret(HidT, sec_hidt, S)
                                    /\ witness(S, T, auth_s, AUTHS')
                                    /\ request(S, T, auth_i, AUTHI')
end role

role soT (T, S: agent,
          P, Idt, HidT: text,
          Ps: message,
          MECC, MULT, H, ADD, ADDP: hash_func,
          SND, RCV: channel(dy))
played_by T def=
local State: nat,
      Ri: text,
      RI, RS, MASK, AUTHS, AUTHI, E, M: message
const sec_ri, sec_idt, sec_hidt, auth_s, auth_i: protocol_id
init State:=1
transition
1. State=1 /\ RCV(RS') =|> State':=3 /\ Ri':=new()
                                     /\ RI':=MECC(Ri', P)
                                     /\ MASK':=MECC(Ri', Ps)
                                     /\ AUTHI':=xor(Idt, H(RS', MASK'))
                                     /\ M':=xor(H(MASK'), HidT)
                                     /\ SND(RI', AUTHI', M')
                                     /\ witness(T, S, auth_i, AUTHI')
                                     /\ secret(Idt, sec_idt, T)
                                     /\ secret(HidT, sec_hidt, T)
                                     /\ secret(Ri', sec_ri, T)
2. State= 3 /\ RCV (AUTHS')
            /\ E'= H(RS, RI, Idt)
            /\ AUTHS'= ADDP(MECC(E',Ps),RS) =|> State':=5 /\ request(T, S, auth_s, AUTHS')
end role

role session(T, S: agent,
             P, Idt, HidT: text,
```

```
            Ps: message,
            MECC,MULT, H, ADD, ADDP: hash_func)
def=
local ST, RT, SS, RSS: channel (dy),
      Xs: text
composition
          serverS(S, T, P, Xs, Idt, HidT, Ps,  MECC, MULT, H, ADD, ADDP, SS, RSS)
          /\ soT(T, S, P, Idt, HidT, Ps,  MECC, MULT, H,  ADD, ADDP, ST, RT)
end role

role environment() def=
const  servers, sot, i: agent,
       mecc, mult, h, add, addp: hash_func,
       p, idt, hidt : text,
       ps: message,
       auth_s, auth_i: protocol_id
intruder_knowledge = {servers, sot, i, mecc, mult, h, add, addp}
composition
          session(servers, sot, p, idt, hidt, ps, mecc, mult, h, add, addp)
          /\ session(servers, i, p, idt, hidt, ps, mecc, mult, h, add, addp)
          /\ session(i, sot, p, idt, hidt, ps, mecc, mult, h, add, addp)
end role
goal
secrecy_of sec_ri, sec_rs, sec_xs, sec_idt, sec_hidt
authentication_on auth_s, auth_i
end goal
environment ()
```

A.3 Protocole d'authentification basé sur l'utilisation de fonctions de hachage

Il existe deux rôles de base S et T qui expliquent l'activité de $Server$ et T_i.

- Il existe deux agents `T` et `S` qui utilisent une fonction de hachage `H`, une fonction de multiplication scalaire `MECC`, un générateur de courbe `P` plus l'identité d'un objet intelligent `T Idt` et le résultat du hachage de son identité `HidT` pour calculer les messages d'authentification `AUTHI` et `AUTHS`.
- La confidentialité (le secret, « secrecy ») de l'identité de l'objet intelligent `T Idt`, du résultat son hachage `HidT` et des nombres aléatoires secrets `Ri`, `Rs` est modélisée à l'aide des prédicats `secret(Idt, sec_idt, T)`, `secret(Idt, sec_idt, S)`, `secret(HidT, sec_hidt, T)`, `secret(HidT, sec_hidt, S)`, `secret(Ri', sec_ri, T)`, `secret(Rs', sec_rs, S)` qui sont gérés par le `protocol_id` : respectivement `sec_idt`, `sec_hidt`, `sec_ri` et `sec_rs`. Les paramètres `Idt`, `HidT` et `Ri`, `Rs` sont gardés secrets pour `T` et `S`.

- L'authentification mutuelle est réalisée via les objectifs `witness` et `request`, c'est-à-dire `witness(T, S, auth_i, AUTHI')`,`request(S, T, auth_i, AUTHI')`,`witness(S, T, auth_s, AUTHS')`,`request(T, S, auth_s, AUTHS')`.`witness(T, S, auth_i, AUTHI')` déclare que l'agent `T` prétend être l'homologue de l'agent `S`, en accord sur la valeur `AUTHI'`. `auth_i` représente l'authentification de `AUTHI'` indiquée dans la section objectif, alors que `request(S, T, auth_i, AUTHI')` déclare que l'agent `S` accepte la valeur `AUTHI'` et s'appuie désormais sur la garantie que l'agent `T` existe et est d'accord avec lui à propos de cette valeur.
- L'environnement contient les constantes globales et la composition d'une ou plusieurs sessions. L'intrus participe en tant que session concrète à l'exécution du protocole.

Listing A.2 – Spécification AVISPA

```
role serverS (S, T :agent,
              P, Idt, HidT: text,
              H, MECC: hash_func,
              SND, RCV : channel(dy))
played_by S def=
local State: nat,
      Rs, K:text,
      RI, RS, M, AUTHS, AUTHI: message
      const sec_rs, sec_idt, sec_hidt, auth_i, auth_s: protocol_id
init State:=0
transition
1. State=0 /\ RCV(start) =|> State':=2 /\ Rs':=new()
                                       /\ RS':=MECC(Rs', P)
                                       /\ secret(Rs', sec_rs, S)
                                       /\ SND(RS')
2. State=2 /\ RCV(RI', AUTHI', M')
           /\ K'= MECC(Rs, RI')
           /\ HidT=xor(M', K')
           /\ AUTHI=H(Idt'. 2. K') =|> State':=4 /\ AUTHS':=H(Idt.1.K')
                                                 /\ SND(AUTHS')
                                                 /\ secret(Idt, sec_idt, S)
                                                 /\ secret(HidT, sec_hidt, S)
                                                 /\ request( S, T, auth_i, AUTHI')
                                                 /\ witness(S, T, auth_s, AUTHS')
end role

role soT(T, S:agent,
         P, Idt, HidT: text,
         H, MECC: hash_func,
         SND, RCV : channel(dy))
played_by T def=
local State: nat,
      Ri, K: text,
      RI, RS, M, AUTHS, AUTHI: message
```

```
const sec_ri, sec_idt, sec_hidt, auth_i, auth_s : protocol_id
init State:=1
transition
1.State=1 /\ RCV(RS') =|> State':=3 /\ Ri':=new()
                                    /\ RI':=MECC(Ri', P)
                                    /\ K':=MECC(Ri', RS')
                                    /\ AUTHI':=H(Idt.2.K')
                                    /\ M':=xor(HidT, K')
                                    /\ SND(RI', AUTHI', M')
                                    /\ secret(Idt, sec_idt, T)
                                    /\ secret(HidT, sec_hidt, T)
                                    /\ secret(Ri', sec_ri, T)
                                    /\ witness(T, S, auth_i, AUTHI')
2.State=3 /\ RCV(AUTHS')
          /\ AUTHS'=H(Idt.1. K) =|> State':=5 /\ request(T, S, auth_s, AUTHS')
end role

role session(T, S: agent,
             P,Idt, HidT: text,
             H, MECC: hash_func)
def=
local ST, RT, SS, RSS: channel(dy)
composition
          serverS(S, T, P, Idt, HidT, H, MECC, SS, RSS)
          /\ soT(T, S, P, Idt, HidT, H, MECC, ST, RT)
end role

role environment() def=
const  servers, sot, i: agent,
       h, mecc: hash_func,
       p, idt, hidt: text,
       auth_i, auth_s: protocol_id
intruder_knowledge= {servers, sot, i, h, mecc}
composition
           session(servers, sot, p, idt, hidt, h, mecc)
           /\ session(servers, i, p, idt, hidt, h, mecc)
           /\ session(i, sot, p, idt, hidt, h, mecc)
end role
goal
secrecy_of sec_ri, sec_rs, sec_idt, sec_hidt
authentication_on auth_i, auth_s
end goal
environment ()
```

A.4 Protocole d'authentification M2M basé sur ECC

Il existe deux rôles de base A et B qui expliquent l'activité de T_i et T_j.

- Il existe deux agents A et B qui utilisent une fonction de hachage H, une fonction de multi-

plication modulaire MULT, une fonction de multiplication scalaire MECC, une fonction d'addition de points ADDP, un générateur de courbe P, et la clé publique du serveur Ps, les clés privées/publiques Xa/Pa, Xb/Pb, et les signatures de Schnorr de la clé publique (Ppa, Za), (Ppb, Zb) pour calculer les messages d'authentification AUTHI et AUTHS.

- La confidentialité (le secret, « secrecy ») des clés privées Xa, Xb et des nombres aléatoires secrets Ra, Rb est modélisée à l'aide des prédicats secret(Xa, sec_xa, A), secret(Xb, sec_xb, B), secret(Ra', sec_ra, A), secret(Rb', sec_rb, B), qui sont gérés par le protocol_id : respectivement sec_xa, sec_xb, sec_ra, sec_rb. Les paramètres (Xa, Ra) et (Xb, Rb) sont gardés secrets pour A et B.
- L'authentification mutuelle est réalisée via les objectifs witness et request, c'est-à-dire witness(A, B, auth_a, AUTHA'), request(B, A, auth_a, AUTHA'), witness(B, A, auth_b, AUTHB'), request(A, B, auth_b, AUTHB'). witness(A, B, auth_a, AUTHA') déclare que l'agent A prétend être l'homologue de l'agent B, en accord sur la valeur AUTHA'. auth_a représente l'authentification de AUTHA' indiquée dans la section objectif, alors que request(B, A, auth_a, AUTHA') déclare que l'agent B accepte la valeur AUTHA' et s'appuie désormais sur la garantie que l'agent A existe et est d'accord avec lui à propos de cette valeur.
- L'environnement contient les constantes globales et la composition d'une ou plusieurs sessions. L'intrus participe en tant que session concrète à l'exécution du protocole.

Listing A.3 – Spécification AVISPA

```
role soA(A,B: agent,
         Xa, P: text,
         Ps, Pa, PPa, Za: message,
         H, MULT, MECC, ADDP, INITVAR: hash_func,
         SND, RCV: channel(dy))
played_by A def=
local State: nat,
      Ra: text,
      R1, R2, AUTHA, AUTHB, Pb, PPb, Zb, INTER: message
const sec_xa, sec_ra, auth_a, auth_b: protocol_id
init State:=0
transition
1. State=0 /\ RCV(start) =|> State':=2 /\ Ra':=new()
                                       /\ R1':=MECC(Ra', Pa)
                                       /\ secret(Xa, sec_xa, A)
                                       /\ secret(Ra', sec_ra, A)
                                       /\ SND(R1', Pa, PPa, Za)

2. State= 2 /\ RCV(R2', AUTHB', Pb', PPb', Zb')
            /\ INTER'=INITVAR
```

```
            /\ INTER'=MECC(H(Pb',PPb'), Ps)
            /\ ADDP(PPb', INTER')=MECC(Zb', P)
            /\ AUTHB' = MECC(MULT(Ra, Xa), R2') =|> State':=4 /\ AUTHA':=MECC(Xa, R2')
                                                              /\ SND(AUTHA')
                                                              /\ request(A, B, auth_b, AUTHB')
                                                              /\ witness(A, B, auth_a, AUTHA')
end role

role soB (B, A: agent,
          Xb, P: text,
          Ps, Pb, PPb, Zb: message,
          H, MULT, MECC, ADDP, INITVAR: hash_func,
          SND, RCV: channel(dy))
played_by B def=
local State: nat,
      Rb: text,
      R2, R1, AUTHB, AUTHA, Pa, PPa, Za, INTER: message
const sec_xb, sec_rb,auth_a, auth_b: protocol_id
init State:=1
transition
1.State=1 /\ RCV(R1', Pa', PPa', Za')
          /\ INTER'=INITVAR
          /\ INTER'=MECC(H(Pa', PPa'), Ps)
          /\ ADDP(PPa', INTER')=MECC(Za', P) =|> State':=3 /\ Rb':=new()
                                                        /\ R2':=MECC(Rb', Pb)
                                                        /\ AUTHB':=MECC(Xb, R1')
                                                        /\ secret(Xb, sec_xb, B)
                                                        /\ secret(Rb', sec_rb, B)
                                                        /\ SND(Pb, R2', AUTHB', PPb, Zb)
                                                        /\ witness(B, A, auth_b, AUTHB')
2.State=3 /\ RCV(AUTHA')
          /\ AUTHA'=MECC(MULT(Rb, Xb), R1') =|> State':=5 /\ request(B, A, auth_a, AUTHA')
end role

role session(A, B: agent,
             P: text,
             Ps: message,
             H, MULT, MECC, ADDP, INITVAR: hash_func)
def=
local SA, RA, SB, RB: channel(dy),
      XA, XB : text,
      Pa, Pb, PPa, Za, PPb, Zb: message
composition
          soA(A, B, P, XA, Ps, Pa, PPa, Za, H, MULT, MECC, ADDP, INITVAR, SA, RA)
          /\ soB(B, A, P, XB, Ps, Pb, PPb, Zb, H, MULT, MECC, ADDP, INITVAR, SB, RB)
end role

role environment() def=
const soa, sob, i: agent,
      pa, pb, pi, ps, ppa, za, ppb, zb: message,
```

```
      h, mult, mecc, addp, initvar: hash_func,
      xa, xb, p: text,
      auth_a, auth_b: protocol_id
intruder_knowledge={soa, sob, i, h, mult, mecc, addp, initvar}
composition
          session(soa, sob, p, ps, h, mult, mecc, addp, initvar)
          /\ session(soa, i, p, ps, h, mult, mecc, addp, initvar)
          /\ session (i, sob, p, ps, h, mult, mecc, addp, initvar)
end role
goal
secrecy_of sec_xa, sec_xb, sec_ra, sec_rb
authentication_on auth_a, auth_b
end goal
environment()
```

A.5 Protocole d'authentification RFID basé sur le hachage

Listing A.4 – Verif Avispa

```
role serverS (S, T :agent,
               H1, H2, H3,CONCAT : function,
               SND, RCV : channel (dy) )

played_by S def=
local State                              : nat ,
             Idti                        : text ,
             Kti,KGI                     : symmetric_key,
             Rs, Rt, MSG  , MSGs               : message
const        sec_kti, sec_kgi, sec_idt, auth_kti : protocol_id

init State:=0

transition

1. State = 0 /\ RCV (start)
             =|> State':=2
               /\ Rs':=new()
              /\SND (Rs')

2. State= 2   /\RCV (MSG')
              /\  MSG'= {CONCAT(Rs,Rt',H1(Kti),Idti)}_KGI
          =|> State':=4
                 /\ request(S, T, auth_kti,Kti)
                 /\ Kti':=H2(Kti)
                 /\MSGs':= H3(Kti',Rt)
```

```
                /\ SND (MSGs')
                /\secret(Idti, sec_idt, {T,S})
                /\secret(Kti, sec_kti, {T, S})
                /\secret(KGI, sec_kgi, {T, S})
                /\ witness(S, T, auth_kti,Kti')
end role


role tagT ( T, S :agent,
            H1, H2, H3,CONCAT : function,
            SND, RCV : channel (dy) )

played_by   T   def=

local State                          : nat ,
          Idti                       : text ,
         Kti,   KGI                  : symmetric_key,
         Rs, Rt, MSG , MSGs          : message
const           sec_kti,sec_kgi, sec_idt, auth_kti : protocol_id

init State:=1

transition
1. State = 1 /\ RCV (Rs')
             =|> State':=3
               /\ Rt':=new()
               /\ MSG':= {CONCAT(Rs',Rt',H1(Kti),Idti')}_KGI
                 /\ SND(MSG')
                 /\secret(Idti', sec_idt, {T,S})
                 /\secret(Kti, sec_kti, {T,S})
                 /\secret(KGI, sec_kgi, {T, S})
                 /\witness(T, S, auth_kti, Kti)

2. State = 3 /\ RCV (MSGs')/\MSGs'=H3(H2(Kti'),Rt) =|> State':=5
                                    /\Kti':=H2(Kti')
                                   /\request(T, S, auth_kti, Kti')
end role


role session ( S, T  :  agent ,
                  H1, H2, H3, CONCAT : function       )
def=
```

```
local ST, RT, SS, RSS : channel (dy)

composition
     serverS ( S, T, H1, H2, H3, CONCAT, SS, RSS)
/\   tagT   ( T, S, H1, H2, H3, CONCAT, ST, RT)

end role

role environment() def=
const  a, b, i : agent ,
       h1, h2, h3, concat : function ,
       sec_kti,sec_kgi, sec_idt, auth_kti : protocol_id
       intruder_knowledge = { a, b, i, h1, h2,h3, concat}

composition
    session (a, b,  h1, h2, h3, concat)
/\  session (a, i,  h1, h2, h3, concat)
/\  session (i, b,  h1, h2, h3, concat)
end role
goal
secrecy_of sec_kti,sec_kgi, sec_idt
authentication_on auth_kti
end goal
environment ()
```

Annexe B

Vérification formelle de protocoles avec ProVerif

B.1 ProVerif

ProVerif est un vérificateur automatique de protocoles cryptographiques basé sur l'approche de la théorie de Horn pour la représentation des protocoles et des intrus. En outre, ProVerif est un outil de vérification automatisé permettant d'analyser les protocoles de sécurité contre des adversaires puissants.

Les protocoles cryptographiques et les objectifs de sécurité associés sont écrits de manière formelle via une variante appliquée du langage Pi-calculus. Cette extension du Pi-calculus permet à ProVerif de gérer de nombreuses primitives de chiffrement symétrique et à clé publique spécifiées sous forme de règles de réécriture ou d'équations. De plus, ProVerif peut implémenter une fonction de hachage à sens unique, des fonctions cryptographiques, et des signatures numériques, etc.

Dans cette annexe, les protocoles proposés dans les chapitres 3 et 4 sont modélisés dans ProVerif et leur évaluation de sécurité est effectuée. Le modèle de vérificateur de ProVerif comprend trois parties :

(1) Déclaration : cette partie définit toutes les variables, noms et canaux de communication, y compris les fonctions de chiffrements utilisées ;

(2) Processus : elle inclut tous les processus et sous-processus du SO et du serveur ;

(3) La partie principale contient le schéma d'investigation. Dans la partie principale, nous simulons une exécution parallèle des processus. Pour vérifier la propriété d'accessibilité, nous définissons les événements de début et de fin du serveur/SO et du SO/TAG.

B.2 Spécification du Script Pi Calculus appliqué à l'amélioration du protocole d'authentification M2C de Jin et al.

Les explication des scripts du Pi Calculus appliqué sont :

- Les secrets `xs`, `idt` et `hidt` sont déclarés comme tels pour l'attaquant en utilisant le mot clé `[private]`. `ch` est un canal public où `SERVERS` et `SOI` échangent leurs messages. Les fonctions de hachages sont modélisées par `H1`, `H2` et `H3`, pour respectivement le hachage de un, deux ou trois éléments. `mult`, `mecc`, `add`, `addp`, `concat` et `xor` représentent respectivement la multiplication modulaire, la multiplication scalaire, l'addition modulaire, l'addition de points, la concatenation et le xor.
- La confidentialité (le secret, « secrecy ») de `idt`, `hidt`, `xs`, `ri` et `rs` est vérifiée avec les requêtes `query attacker(idt)`, `query attacker(hidt)`, `query attacker(xs)`, `query attacker(new ri)` et `query attacker(new rs)`.
- L'authentification mutuelle entre T_i et le $Server$ est modélisée avec la définition de quatre évènements qui sont liés aux sous-processus `SOI` et `SERVERS` et aux requêtes suivantes :

```
event beginT(bitstring).
event endT(bitstring).
event beginS(bitstring).
event endS(bitstring).

query x: bitstring; inj-event(beginS(x)) ==> inj-event(beginT(x)).
query x: bitstring; inj-event(endT(x)) ==> inj-event(endS(x)).
query x: bitstring, y:bitstring; inj-event(endS(y)) ==>
inj-event(beginS(x)) && inj-event(beginT(x)).
```

- Dans le processus principal, les sous-processus `SERVERS` et `SOI` s'exécutent en parallèle. `!` indique un nombre illimité de processus :

```
process
((!SERVERS) | (!SOI))
```

Listing B.1 – Spécification ProVerif

```
(* Jin et al.'s protocol improvements *)

free ch: channel.
free xs, idt, hidt: bitstring [private].
free ps, p: bitstring.
fun H1(bitstring):bitstring.
fun H2(bitstring, bitstring): bitstring.
fun H3(bitstring, bitstring, bitstring): bitstring.
fun mult(bitstring, bitstring): bitstring.
fun mecc(bitstring, bitstring): bitstring.
```

```
fun add(bitstring, bitstring): bitstring.
fun addp(bitstring, bitstring): bitstring.
fun xor(bitstring,bitstring): bitstring.
equation forall x: bitstring, y: bitstring; xor(xor(x, y), y)=x.

(* Queries *)
(* verify secrets *)
query attacker(xs).
query attacker(idt).
query attacker(hidt).
query attacker(new ri).
query attacker(new rs).
(* verify authentication *)
event beginT(bitstring).
event endT(bitstring).
event beginS(bitstring).
event endS(bitstring).
query x: bitstring; inj-event(beginS(x)) ==> inj-event(beginT(x)).
query x: bitstring; inj-event(endT(x)) ==> inj-event(endS(x)).
query x: bitstring, y:bitstring; inj-event(endS(y)) ==>
inj-event(beginS(x))&&inj-event(beginT(x)).

(* Role of the Server S *)
let SERVERS=
new rs: bitstring;
let RS=mecc(rs, p) in
out(ch, (RS));
in(ch,(m2:bitstring, m3:bitstring, m4:bitstring));
let Mask=mecc(xs, m2) in
let hidtprime=xor(m4, H1(Mask)) in
let idtprime=xor(m3, H2(RS, Mask)) in
if idt=idtprime then
let e= H3(RS, m2, idt) in
let Auths= mecc(add(mult(xs, e), rs), p) in
event beginS(Auths);
out (ch, Auths);
event endS(Auths).

(* Role of the Smart Object I *)
let SOI=
in(ch,(m1: bitstring));
new ri: bitstring;
let RI=mecc(ri, p) in
let Mask=mecc(ri, ps) in
let Authi=xor(idt, H2(m1, Mask)) in
let M=xor(H1(Mask), hidt) in
event beginT(Authi);
out(ch, (RI, Authi, M));
in(ch, m5: bitstring);
let e=H3(m1, RI, idt) in
```

```
if m5=addp(mecc(e, ps), m1)then
event endT(Authi).

(* Start process *)
process
((* Launch an unbounded number of sessions of the smart object AND server *)
   (!SERVERS) | (!SOI)
```

B.3 Résultats ProVerif de l'amélioration du protocole d'authentification M2C de Jin et al.

Listing B.2 – Résultats ProVerif

```
-- Query not attacker(xs[])
Completing...
Starting query not attacker(xs[])
RESULT not attacker(xs[]) is true.
-- Query not attacker(idt[])
Completing...
Starting query not attacker(idt[])
RESULT not attacker(idt[]) is true.
-- Query not attacker(hidt[])
Completing...
Starting query not attacker(hidt[])
RESULT not attacker(hidt[]) is true.
-- Query not attacker(ri[m1 = v_2207,!1 = v_2208])
Completing...
Starting query not attacker(ri[m1 = v_2207,!1 = v_2208])
RESULT not attacker(ri[m1 = v_2207,!1 = v_2208]) is true.
-- Query not attacker(rs[!1 = v_2765])
Completing...
Starting query not attacker(rs[!1 = v_2765])
RESULT not attacker(rs[!1 = v_2765]) is true.
-- Query inj-event(beginS(x_30)) ==> inj-event(beginT(x_30))
Completing...
Starting query inj-event(beginS(x_30)) ==> inj-event(beginT(x_30))
RESULT inj-event(beginS(x_30)) ==> inj-event(beginT(x_30)) is true.
-- Query inj-event(endT(x_31)) ==> inj-event(endS(x_31))
Completing...
Starting query inj-event(endT(x_31)) ==> inj-event(endS(x_31))
RESULT inj-event(endT(x_31)) ==> inj-event(endS(x_31)) is true.
-- Query inj-event(endS(y_33)) ==> (inj-event(beginS(x_32)) && inj-event(beginT(x_32)))
Completing...
Starting query inj-event(endS(y_33)) ==> (inj-event(beginS(x_32)) &&
inj-event(beginT(x_32)))
RESULT inj-event(endS(y_33)) ==> (inj-event(beginS(x_32)) &&
inj-event(beginT(x_32))) is true.
```

B.4 Spécification du Script Pi Calculus appliqué au protocole d'authentification basé sur l'utilisation de fonctions de hachage

Les explication des scripts du Pi Calculus appliqué sont :

- Les secrets `idt` et `hidt` sont déclarés comme tels pour l'attaquant en utilisant le mot clé `[private]`. `ch` est un canal public où `SERVERS` et `SOI` échangent leurs messages. `h`, `mecc`, `concat` et `xor` représentent respectivement la fonction de hachage, multiplication modulaire, la concatenation et le xor.
- La confidentialité (le secret, « secrecy ») de `idt`, `hidt`, `ri` et `rs` est vérifiée avec les requêtes `query attacker(idt)`, `query attacker(hidt)`, `query attacker(new ri)` et `query attacker(new rs)`.
- L'authentification mutuelle entre T_i et le $Server$ est modélisée avec la définition de quatre évènements qui sont liés aux sous-processus `SOI` et `SERVERS` et aux requêtes suivantes :

```
event beginT(bitstring).
event endT(bitstring).
event beginS(bitstring).
event endS(bitstring).

query x: bitstring; inj-event(beginS(x)) ==> inj-event(beginT(x)).
query x: bitstring; inj-event(endT(x)) ==> inj-event(endS(x)).
query x: bitstring, y:bitstring; inj-event(endS(y)) ==> inj-event(beginS(x)) &&
                                                        inj-event(beginT(x)).
```

- Dans le processus principal, les sous-processus `SERVERS` et `SOI` s'exécutent en parallèle. `!` indique un nombre illimité de processus :

```
process
((!SERVERS) | (!SOI))
```

Listing B.3 – Spécification ProVerif

```
    (* Hash-based authentication protocol *)
free ch: channel.
free idt, hidt: bitstring [private].
free ps, p: bitstring.
const v1,v2: bitstring.
fun h (bitstring): bitstring [data].
fun mecc (bitstring, bitstring):bitstring.
fun concat (bitstring, bitstring):bitstring.
fun xor (bitstring, bitstring): bitstring.
equation forall x:bitstring, y: bitstring; xor(xor(x,y),y)=x.

(* Queries *)
(* verify secrets*)
```

```
query attacker(idt).
query attacker(hidt).
query attacker(new ri).
query attacker(new rs).
(* verify authentication *)
event beginT(bitstring).
event endT(bitstring).
event beginS(bitstring).
event endS(bitstring).
query x: bitstring; inj-event(beginS(x)) ==> inj-event(beginT(x)).
query x: bitstring; inj-event(endT(x)) ==> inj-event(endS(x)).
query x: bitstring, y:bitstring; inj-event(endS(y)) ==>
inj-event(beginS(x)) && inj-event(beginT(x)).

(* Role of the Server S *)
let SERVERS =
new rs:bitstring;
let RS=mecc(rs, p) in
out(ch, (RS));
in(ch,(m2: bitstring, m3: bitstring, m4: bitstring));
let KS=mecc(rs, m2) in
let hidtprime = xor(m4, KS) in
if hidt=hidtprime then
if m3=h(concat(idt, concat(v2, KS))) then
let AUTHS=h(concat(idt, concat(v1, KS))) in
event beginS(AUTHS);
out (ch, AUTHS);
event endS(AUTHS).

(* Role of the smart object I *)
let SOI =
in(ch,(m1: bitstring));
new ri: bitstring;
let RI=mecc(ri, p) in
let KI=mecc(ri, m1) in
let h(=idt)=hidt in
let M=xor(hidt, KI) in
let AUTHI=h(concat(idt, concat(v2, KI))) in
event beginT(AUTHI);
out(ch,(RI, AUTHI, M));
in(ch, m5: bitstring);
if m5= h(concat(idt, concat(v1, KI)))then
event endT(AUTHI).

(* Start process *)
process
        ((* Launch an unbounded number of sessions of the TAG AND SERVER*)
          (!SERVERS) | (!SOI))
```

B.5 Résultats ProVerif du protocole d'authentification basé sur l'utilisation de fonctions de hachage

Listing B.4 – Résultats ProVerif

```
-- Query not attacker(idt[])
Completing...
Starting query not attacker(idt[])
RESULT not attacker(idt[]) is true.
-- Query not attacker(hidt[])
Completing...
Starting query not attacker(hidt[])
RESULT not attacker(hidt[]) is true.
-- Query not attacker(ri[m1 = v_766,!1 = v_767])
Completing...
Starting query not attacker(ri[m1 = v_766,!1 = v_767])
RESULT not attacker(ri[m1 = v_766,!1 = v_767]) is true.
-- Query not attacker(rs[!1 = v_1021])
Completing...
Starting query not attacker(rs[!1 = v_1021])
RESULT not attacker(rs[!1 = v_1021]) is true.
-- Query inj-event(beginS(x_28)) ==> inj-event(beginT(x_28))
Completing...
Starting query inj-event(beginS(x_28)) ==> inj-event(beginT(x_28))
RESULT inj-event(beginS(x_28)) ==> inj-event(beginT(x_28)) is true.
-- Query inj-event(endT(x_29)) ==> inj-event(endS(x_29))
Completing...
Starting query inj-event(endT(x_29)) ==> inj-event(endS(x_29))
RESULT inj-event(endT(x_29)) ==> inj-event(endS(x_29)) is true.
-- Query inj-event(endS(y_31)) ==> (inj-event(beginS(x_30)) && inj-event(beginT(x_30)))
Completing...
Starting query inj-event(endS(y_31)) ==> (inj-event(beginS(x_30)) &&
inj-event(beginT(x_30)))
RESULT inj-event(endS(y_31)) ==> (inj-event(beginS(x_30)) &&
inj-event(beginT(x_30))) is true.
```

B.6 Spécification du Script Pi Calculus appliqué au protocole d'authentification M2M basé sur ECC

Les explication des scripts du Pi Calculus appliqué sont :

- Les secrets `xa`, `xb` sont déclarés comme tels pour l'attaquant en utilisant le mot clé `[private]`. `ch` est un canal public où `SOA` and `SOB` échangent leurs messages. `h`, `mult`, `mecc`, `addp` représentent respectivement la fonction de hachage, la multiplication scalaire, modulaire, la multiplication scalaire et l'addition de points.
- La confidentialité (le secret, « secrecy ») de `xa`, `xb`, `ra` et `rb` est vérifiée avec les requêtes

`query attacker(xa)`, `query attacker(xb)`, `query attacker(new ra)` et `query attacker(new rb)`.

- L'authentification mutuelle entre T_i et T_j est modélisée avec la définition de quatre évènements qui sont liés aux sous-processus `SOA` et `SOB` et aux requêtes suivantes :

Listing B.5 – Spécification ProVerif

```
    (* ECC based authentication protocol *)

free ch: channel.
free xa, xb: bitstring [private].
free pa, pb, ps, p, ppa, ppb, za, zb: bitstring.
fun h (bitstring, bitstring): bitstring.
fun mult (bitstring, bitstring):bitstring.
fun mecc (bitstring, bitstring):bitstring.
fun addp (bitstring, bitstring):bitstring.

(* Queries *)
(* verify secrets *)
query attacker(xa).
query attacker(xb).
query attacker(new ra).
query attacker(new rb).
(* verify authentication *)
event beginA(bitstring).
event endA(bitstring).
event beginB(bitstring).
event endB(bitstring).
query x: bitstring; inj-event(beginB(x)) ==> inj-event(beginA(x)).
query x: bitstring; inj-event(endA(x)) ==> inj-event(endB(x)).
query x: bitstring, y:bitstring; inj-event(endA(y)) ==>
inj-event(beginA(x)) && inj-event(beginB(x)).

(* Role of the smart object A *)
let SOA=
new ra: bitstring;
let RA=mecc(ra, pa) in
out(ch, (RA, pa, ppa, za));
in(ch,(m5: bitstring, m6: bitstring, m7: bitstring, m8: bitstring, m9: bitstring));
if addp(mecc(h(m7, m8), ps), m8)= mecc(m9, p) then
if m6=mecc( mult(ra, xa), m7) then
let AUTHA=mecc(xa, m5) in
event beginA(AUTHA);
out (ch, AUTHA);
event endA(AUTHA).

(* Role of the smart object B *)
let SOB=
in(ch,(m1: bitstring, m2: bitstring, m3: bitstring, m4: bitstring));
```

```
new rb: bitstring;
if addp(mecc(h(m2, m3), ps), m3)=mecc(m4, p) then
let RB= mecc(rb, pb) in
let AUTHB=mecc(xb, m1) in
event beginB(AUTHB);
out(ch,(RB, AUTHB, pb, ppb, zb));
in(ch, m10: bitstring);
if m10= mecc( mult(rb, xb),m2) then
event endB(AUTHB).

(* Start process *)
process((* Launch an unbounded number of sessions of the SOA AND SOB *)
          (!SOA) | (!SOB))
```

B.7 Résultats ProVerif du protocole d'authentification M2M basé sur ECC

Listing B.6 – Résultats ProVerif

```
-- Query not attacker(xa[])
Completing...
Starting query not attacker(xa[])
RESULT not attacker(xa[]) is true.
-- Query not attacker(xb[])
Completing...
Starting query not attacker(xb[])
RESULT not attacker(xb[]) is true.
-- Query not attacker(ra[!1 = v_931])
Completing...
Starting query not attacker(ra[!1 = v_931])
RESULT not attacker(ra[!1 = v_931]) is true.
-- Query not attacker(rb[m4 = v_1234,m3 = v_1235,m2 = v_1236,m1 = v_1237,!1 = v_1238])
Completing...
Starting query not attacker(rb[m4 = v_1234,m3 = v_1235,m2 = v_1236,m1 = v_1237,
!1 = v_1238])
RESULT not attacker(rb[m4 = v_1234,m3 = v_1235,m2 = v_1236,m1 = v_1237,!1 = v_1238])
is true.
-- Query inj-event(beginB(x)) ==> inj-event(beginA(x))
Completing...
Starting query inj-event(beginB(x)) ==> inj-event(beginA(x))
RESULT inj-event(beginB(x)) ==> inj-event(beginA(x)) is true.
-- Query inj-event(endA(x_12)) ==> inj-event(endB(x_12))
Completing...
Starting query inj-event(endA(x_12)) ==> inj-event(endB(x_12))
RESULT inj-event(endA(x_12)) ==> inj-event(endB(x_12)) is true.
-- Query inj-event(endA(y)) ==> (inj-event(beginA(x_13)) && inj-event(beginB(x_13)))
Completing...
Starting query inj-event(endA(y)) ==> (inj-event(beginA(x_13)) &&
```

```
inj-event(beginB(x_13)))
RESULT inj-event(endA(y)) ==> (inj-event(beginA(x_13)) &&
inj-event(beginB(x_13))) is true.
```

B.8 Spécification du Script Pi Calculus appliqué au protocole d'authentification RFID basé sur le hachage

Listing B.7 – Verif ProVerif

```
(*
Group based authentication protocol
*)

type skey.

free ch: channel.
free idt: bitstring [private].
free Rt: bitstring.
free kti: skey [private].
free kgi: skey [private].
fun h1 (skey): bitstring.
fun h2 (skey): bitstring.
fun h3 (bitstring): bitstring.
fun concat (bitstring, bitstring, bitstring, bitstring):bitstring.
fun concat2 (bitstring, bitstring):bitstring.
fun enc (bitstring, skey): bitstring.

(* Queries *)

(* verifier les secrets*)

query attacker(kti).
query attacker(kgi).
query attacker(idt).



(*verifier l'authentification*)

event beginT(bitstring).
event endT(bitstring).
event beginS(bitstring).
```

```
event endS(bitstring).

query x: bitstring; inj-event(beginS(x)) ==> inj-event(beginT(x)).
query x: bitstring; inj-event(endT(x)) ==> inj-event(endS(x)).
query x: bitstring, y:bitstring; inj-event(endS(y)) ==>
inj-event(beginS(x)) && inj-event(beginT(x)).

(* Role of the Server S*)

let SERVERS =
new Rs:bitstring;
out(ch,Rs);
in(ch,(m2: bitstring));
if m2=enc(concat(Rs,Rt,h1(kti),idt),kgi) then
let ktiplus=h2(kti) in
event beginS(m2);
out(ch,h3(concat2(ktiplus,Rt)));
event endS(m2).

(* Role of the Tag I *)

let TAGI =
in(ch,(m1: bitstring));
new Rt:bitstring;
let msg= enc(concat(m1,Rt,h1(kti),idt),kgi) in
event beginT(msg);
out(ch, msg);
in(ch,(m3: bitstring));
if m3=h3(concat2(h2(kti),Rt)) then
let ktiplus=h2(kti) in
event endT(msg).

(* Start process *)

process
       (

          (* Launch an unbounded number of sessions of the TAG AND SERVER *)
          (!SERVERS) | (!TAGI)

        )
```

B.9 Résultats ProVerif du protocole d'authentification RFID basé sur le hachage

Listing B.8 – Verif ProVerif Results

```
-- Query not attacker(kti[])
Completing...
Starting query not attacker(kti[])
RESULT not attacker(kti[]) is true.
-- Query not attacker(kgi[])
Completing...
Starting query not attacker(kgi[])
RESULT not attacker(kgi[]) is true.
-- Query not attacker(idt[])
Completing...
Starting query not attacker(idt[])
RESULT not attacker(idt[]) is true.
-- Query inj-event(beginS(x)) ==> inj-event(beginT(x))
Completing...
Starting query inj-event(beginS(x)) ==> inj-event(beginT(x))
RESULT inj-event(beginS(x)) ==> inj-event(beginT(x)) is true.
-- Query inj-event(endT(x_12)) ==> inj-event(endS(x_12))
Completing...
Starting query inj-event(endT(x_12)) ==> inj-event(endS(x_12))
RESULT inj-event(endT(x_12)) ==> inj-event(endS(x_12)) is true.
-- Query inj-event(endS(y)) ==> (inj-event(beginS(x_13)) &&
inj-event(beginT(x_13)))
Completing...
Starting query inj-event(endS(y)) ==> (inj-event(beginS(x_13))
&& inj-event(beginT(x_13)))
RESULT inj-event(endS(y)) ==> (inj-event(beginS(x_13))
&& inj-event(beginT(x_13))) is true.
```