UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
ALI BOUCHAREB

Développement d'une architecture flexible pour la gestion et la sécurité des transactions et flux de données pour les objets connectés dans le contexte de la maison intelligente

Avril 2023

CE MÉMOIRE A ÉTÉ ÉVALUÉ PAR

UN JURY COMPOSÉ DE

▪ **Boucif Amar Bensaber**, directeur de recherche.

Professeur au département de Mathématiques et Informatique

Université du Québec à Trois-Rivières.

▪ **Ismail Biskri**, évaluateur.

Professeur au département de Mathématiques et Informatique

Université du Québec à Trois-Rivières.

▪ **Mhamed Mesfioui**, évaluateur.

Professeur au département de Mathématiques et Informatique

Université du Québec à Trois-Rivières.

REMERCIEMENTS

Je voudrais tout d'abord adresser toute ma gratitude au directeur de ce mémoire, le Professeur Boucif Amar Bensaber, pour le temps qu'il a consacré à m'apporter les outils méthodologiques indispensables à la conduite de cette recherche. Pour sa patience, sa disponibilité et surtout pour sa confiance.

Avec beaucoup d'égard, je remercie les professeurs Ismail Biskri et Mhamed Mesfioui d'avoir accepté d'évaluer mon travail.

Je désire aussi remercier les professeurs et tous mes collègues du département de mathématiques et d'informatique, qui m'ont fourni les outils nécessaires à la réussite de mes études.

Enfin, mes vifs remerciements vont à ma famille qui durant l'ensemble de mon parcours m'a soutenue.

RÉSUMÉ

Le développement d'Internet vise à intégrer le monde virtuel d'Internet avec le monde physique à travers l'Internet des objets (IoT) afin d'assurer une meilleure accessibilité et une meilleure exploitation des ressources du monde réel. Avec le développement rapide de l'Internet, il y a un grand nombre d'objets connectés, d'appareils et d'applications dans la société. Il est donc urgent de comprendre les risques associés aux flux de données et aux transactions effectuées entre objets connectés principalement en matière de sécurité et de performance.

À travers un tel paradigme, les progrès récents des technologies de réseaux sans fil entrainent la diffusion de nouvelles applications de l'IoT comprenant la surveillance, la sécurité, la santé, les maisons et les villes intelligentes ainsi que les systèmes de logistique et de transportation intelligents. Dans ce contexte, notre travail se concentre sur la cybersécurité des objets connectés dans un environnement domestique. Nous avons examiné une variété de défis de sécurité pertinents pour la maison intelligente dans la littérature sur l'IoT et nous avons mis en place une maison intelligente fiable en appliquant une architecture flexible pour gérer les transactions et les flux de données pour les objets intelligents de la maison. Notre système aidera à surmonter un certain nombre d'attaques dirigées contre les réseaux domestiques intelligents.

Le système proposé constitue un environnement de maison intelligente basé sur les modèles de réseau centralisé et décentralisé, où nous appliquerons certains protocoles au niveau de la couche « application » pour prendre en charge la gestion du flux de réseau et réduire le coût de calcul des objets connectés à la maison. Nous appliquons également certains protocoles de sécurité à la couche de base pour prendre en charge les vulnérabilités liées aux systèmes « machine to machine (M2M) » et empêcher certains types d'attaques.

Les résultats expérimentaux montrent que le schéma que nous proposons produit un système de sécurité qui protège les maisons intelligentes tout en réduisant le nombre d'attaques d'écoute et de programmes malveillants qui ciblent les objets connectés dans une maison. Nous réduisons également le coût de calcul pour les objets connectés à faible budget énergétique dans le contexte de la maison intelligente.

ABSTRACT

The development of the Internet aims to integrate the virtual world of the Internet with the physical world through the Internet of Things (IoT) to ensure better accessibility and exploitation of real world resources. With the rapid development of the Internet, there are a large number of connected objects, devices and applications in society. Therefore, there is an urgent need to understand the security and performance risks associated with data flows and transactions between connected objects.

Through such a paradigm, recent progress in wireless network technologies is driving the diffusion of new IoT applications including surveillance, security, health, smart homes and cities, and smart logistics and transportation systems. In this context, our work focuses on the cybersecurity of connected objects in a home environment. We have examined a variety of security challenges relevant to the smart home in the IoT literature and have implemented a reliable smart home by applying a flexible architecture to manage transactions and data flows in the IoT domain at home. Our system will help overcome a number of attacks directed against smart home networks.

The proposed system constitutes a smart home environment based on centralized and decentralized network models, where we will apply some protocols to the application layer in order to support the network flow management and to reduce the computational cost of connected objects in the home. We also apply some security protocols to the base layer to support the vulnerabilities related to machine-to-machine (M2M) systems and prevent some types of attacks.

Experimental results show that our proposed scheme produces a security system that can protect smart homes by reducing the number of eavesdropping attacks and malware that target connected objects in a home. We also reduce the computational cost for low-energy connected objects in the smart home context.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

LISTE DES HISTOGRAMMES

LISTE DES ABRÉVIATIONS

6LowPAN	IPv6 Low power Wireless Personal Area Networks
6ED	6LoWPAN End Device
6LBR	6LoWPAN Border Router
6LR	6LoWPAN Router
AES	Advanced Encryption Standard
CoAP	The Constrained Application Protocol
CSMA	Carrier Sense Multiple Access
DOS	DeniaI Of service
DTLS	Datagram Transport Layer Security
ECC	Elliptic-curve cryptography
ECDH	Elliptic-curve Diffie–Hellman
HGW	Home Gateway
HMAC	Hash-Based Message Authentication Code
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of things
M2M	Machine to Machine
MAC	Medium Access Control
MQTT	Message Queuing Telemetry Transport
P2P	Peer to Peer
PHY	Physical-Layer

RFID	Radio Frequency IDentification
RPL	Routing Protocol for Low-Power
RSA	Rivest–Shamir–Adleman
SGW	Service Gateway
SQL	Structured Query Language
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UIT	Union internationale des télécommunications
WSN	Wireless sensor networ

CHAPITRE 1 INTRODUCTION GÉNÉRALE

Le terme « Internet des objets (IoT) » a été mentionné pour la première fois en 1999 par Kevin Ashton et il est devenu aujourd'hui l'un des termes les plus courants que nous rencontrons dans la communauté des chercheurs. Le contrôle à distance des équipements électriques et électroniques (objets) est un concept bien connu depuis le début des années 1990. John Romkey a créé le premier appareil connecté à Internet [1], soit un grille-pain qui peut être allumé et éteint à partir d'Internet, mais qui n'a pas encore été déployé à plein régime dans nos sociétés.

Aujourd'hui, l'IoT est devenu l'un des outils les plus puissants comprenant une variété de communication et de technologies telles que la « Radio Frequency Identification (RFID) », le Bluetooth, le Wi-Fi, le Zigbee, etc., ce qui a rendu son utilisation large dans de nombreux domaines. L'IoT connecte des objets de différents environnements en un seul grand réseau basé sur le protocole Internet et constitue la base du développement des environnements dits intelligents tels que les soins de santé intelligents, les transports intelligents, les maisons intelligentes et les usines, ou même, les villes. Ces objets font référence à tout ce qui nous entoure, allant d'une petite montre portable à un avion géant, et qui permet aux utilisateurs de gérer et d'optimiser les équipements électroniques ainsi qu'électriques à l'aide d'Internet.

L'Internet des objets (IoT) est défini par certains chercheurs comme une collection d'objets intégrés à l'électronique, aux logiciels, aux capteurs et aux actionneurs ainsi que connectés via Internet pour collecter et échanger des données entre eux [2]. Cette dernière définition n'est pas considérée comme une définition standard à l'IoT, car il s'agit d'un domaine tellement évolutif que nous ne savons pas ce qu'il couvrira dans un avenir proche.

La puissance de l'IoT réside à la fois dans le monde physique et dans le monde virtuel. Les données sont numérisées, puis les informations numériques sont envoyées sur le réseau au contrôleur distant pour allumer ou éteindre une machine selon le scénario [3]. L'IoT évolue vers une plate-forme de prise de décision initiée par des capteurs, pilotée par l'actionnement et basée sur l'intelligence artificielle.

Les progrès récents des technologies de l'IoT évoluent vers une plate-forme utilisée de manière interchangeable avec les systèmes « machine à machine (M2M) », où une communication entre des petits appareils et peu coûteux s'effectue sans intervention humaine et où plusieurs dispositifs hétérogènes sont reliés entre eux et exploités pour prendre en charge de tels types de décisions.

La cybersécurité au niveau de la couche inférieure (physique) de la communication M2M dans une maison intelligente est un problème complexe et à multiples facettes. L'énergie est considérée comme l'un des défis les plus importants pour les objets connectés, car ces derniers sont indépendants des sources d'alimentation et utilisent des ressources d'énergie limitée (batterie). Donc, le facteur de gestion d'énergie est très important. Plusieurs recherches se sont concentrées sur l'optimisation des ressources et du taux de calcul pour ces objets.

L'IoT introduit de nouveaux risques et défis de sécurité pour les appareils eux-mêmes, pour leurs plates-formes et systèmes d'exploitation, pour leurs communications M2M ainsi que pour les systèmes auxquels ils sont connectés. Les grandes organisations traitent la sécurité comme un risque à prendre en compte avec tous les autres risques qu'elles gèrent. Les technologies de sécurité seront nécessaires pour protéger les appareils ainsi que les plates-formes de l'IoT des attaques d'informations et de la falsification physique telles que l'usurpation d'identité ou les attaques de l'homme au milieu (« man in middle ») afin de chiffrer leurs communications et de relever de nouveaux défis.

Pour surpasser les risques de la cybercriminalité et les défis liés aux architectures de l'IoT ainsi qu'aux communications M2M dans une maison intelligente, plusieurs auteurs ont proposé différentes solutions de sécurité basées sur plusieurs architectures de communication de l'IoT. La plupart des solutions proposées dans la littérature de l'IoT assurent l'intégrité des données en utilisant des solutions décentralisées à l'aide de la « Blockchain ». D'autres propositions assurent la confidentialité en utilisant des solutions centralisées. En effet, la plupart des solutions proposées dans la littérature de l'IoT ne prennent pas en considération la non-répudiation, l'anonymat ainsi que l'authentification pour les systèmes M2M et ne traitent pas le coût de calcul ainsi que la capacité de stockage pour les objets à faible puissance d'une façon optimale. Pour pallier ces lacunes, nous proposons, dans ce travail, un schéma qui fait partie de l'architecture décentralisée, soit « Peer to Peer (P2P) », pour les communications M2M et un schéma de l'architecture

centralisée qui utilise plusieurs Courtiers de file d'attente des messages Transport de télémétrie « Message Queuing Telemetry Transport (MQTT) Broker ». Ces derniers sont renforcés par l'utilisation de la cryptographie elliptique pour la génération dynamique des clés de chiffrement afin d'améliorer la sécurité de l'IoT dans les environnements domestiques, d'empêcher certains types d'attaques d'écoute liées aux communications M2M et de réduire le coût de calcul pour les objets domestiques.

Pour valider l'efficacité du notre protocole, une évaluation formelle est réalisée par le vérificateur automatique de protocole cryptographique « ProVerif ». Cet outil est caractérisé par sa capacité de gestion des primitives cryptographiques (symétrique et asymétrique). Aussi, cet outil permet de gérer un nombre illimité de sessions de protocole en parallèle. Nous avons évalué également les performances de notre modèle en termes de consommation d'énergie et de trafic généré à l'aide du simulateur « Contiki Cooja », un système d'exploitation libre de droits (« open source ») spécialement conçu pour l'IoT. Les résultats de modélisation ont montré que notre protocole est efficace en termes de non-répudiation, d'anonymat, d'authentification ainsi que de gestion de confiance. Les résultats de simulation ont montré que notre protocole est optimal en termes de consommation d'énergie par rapport aux architectures proposées dans la littérature de l'IoT.

Ainsi, le reste de ce mémoire est divisé en différents chapitres. En effet, le deuxième chapitre présente les généralités et les caractéristiques de l'Internet des objets et des réseaux domestiques. Le troisième chapitre présente quelques travaux issus de la littérature portant sur le problème de la sécurité de l'IoT. Le quatrième chapitre présente la modélisation de notre architecture proposée ainsi que les différents scénarios d'attaques à l'aide de l'outil de vérification automatique « Proverif ». Le cinquième chapitre présente l'évaluation des performances de notre architecture à l'aide du simulateur « Contiki Cooja » ainsi que les améliorations réalisées. Finalement, le dernier chapitre présente une conclusion générale et propose les futurs travaux.

CHAPITRE 2 INTERNET DES OBJETS ET RÉSEAUX DOMESTIQUES : GÉNÉRALITÉS ET CARACTÉRISTIQUES

Dans ce chapitre, nous présenterons plusieurs notions et définitions en lien avec l'Internet des objets, leur sécurité, leur application, la norme « Institute of Electrical and Electronics Engineers (IEEE) » 802, l'architecture de ce type de réseaux, les modes de communications entre les différentes entités du réseau, les services issus de ces réseaux, les mécanismes et les concepts de sécurité des réseaux IoT ainsi que les outils cryptographiques utilisés dans ce travail.

.1 Internet des objets

Selon l'Union internationale des télécommunications (UIT), l'Internet des objets est défini comme une « infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution » [4]. L'IoT peut être considéré comme un concept qui a un impact majeur sur la technologie et la société. En tirant parti des capacités d'identification, de collecte de données, de traitement et de communication, l'IoT tire le meilleur parti des objets pour servir toutes sortes d'applications, tout en garantissant le respect des exigences de sécurité et de confidentialité.

.2 Objets connectés

Dans l'Internet des objets, un objet connecté est une entité physique ou virtuelle capable d'échanger diverses informations à un autre objet ou à Internet. Les données peuvent être collectées, transmises via différents types de connexions et, dans certains cas, traités pour aider à la prise de décision ou initier une action. Dans l'IoT, il existe deux types d'objets.

Les objets physiques appartiennent au monde physique. Ces objets peuvent être détectés, contrôlés et connectés. L'environnement qui nous entoure, les robots industriels, les biens et les équipements électriques sont tous des exemples d'objets physiques.

Les objets virtuels appartiennent au monde de l'information. Ils peuvent être stockés, traités et récupérés. Ces objets sont, par exemple, des contenus multimédias ou des logiciels.

.3 Modèle de référence de l'IoT

La Figure 1 présente le modèle de référence de l'IoT proposé par l'UIT. Ce modèle comprend quatre couches auxquelles sont associées des capacités de gestion et de sécurité.

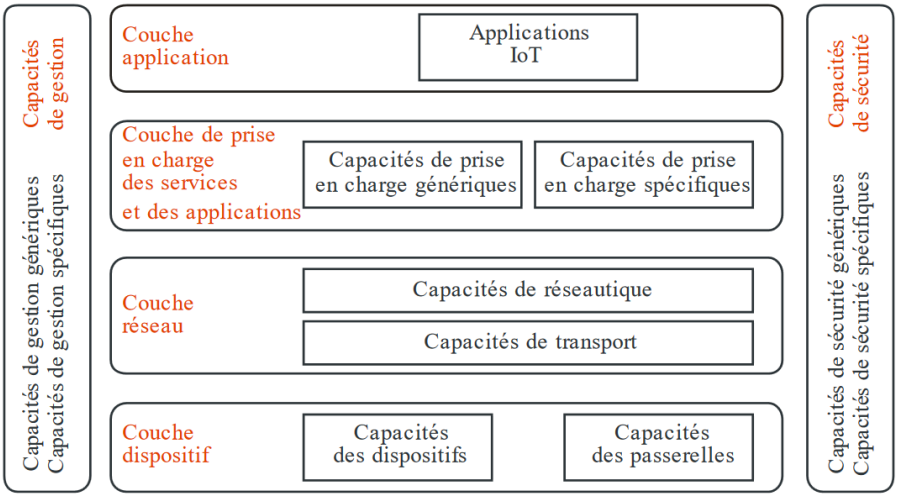


Figure 1: Modèle de référence de l'IoT

.3.1 Couche « application »

La couche « application » est la couche supérieure du modèle de référence de l'IoT, où les applications et l'ensemble des règles définissent les accords de niveau de service de l'IoT.

.3.2 Couche de prise en charge des services et des applications

La couche de prise en charge des services et des applications contient deux ensembles de capacités suivants :

– Capacités de prise en charge génériques : il s'agit de capacités communes qui peuvent être utilisées dans diverses applications de l'IoT telles que la capacité de traitement ou de stockage de données ;

– Capacités de prise en charge spécifiques : il s'agit de capacités particulières répondant aux besoins d'applications diversifiées. Elles peuvent être constituées de différents ensembles de capacités bien précises afin d'assurer des fonctions de prise en charge différentes pour des applications distinctes de l'IoT.

.3.3 Couche « réseau »

Cette couche comprend deux types de capacités :

– Des capacités de réseautique qui fournissent des fonctions de contrôle liées à la connectivité au réseau telles que le contrôle d'accès, le contrôle des ressources de transport, la gestion de la mobilité ou l'authentification, l'autorisation et la comptabilité ;

– Des capacités de transport afin d'assurer la connectivité nécessaire pour le transport des informations propres à chaque service ou à chaque application de l'IoT.

.3.4 Couche « dispositif »

Cette couche contient les deux ensembles de capacités suivants :

– Capacités des dispositifs : les dispositifs sont en mesure de collecter des informations et de les télécharger sur le réseau de communication de manière directe ou indirecte ;

– Capacités des passerelles : elles prennent en charge des dispositifs connectés à l'aide de différentes technologies filaires ou hertziennes, par exemple à l'aide d'un bus gestionnaire de réseau de communication, du protocole ZigBee, du Bluetooth ou du Wi-Fi.

.4 Normes et standards de communication

Les standards de communication sont indispensables pour mettre en œuvre la communication entre les différentes entités dans les réseaux de l'IoT. Une liste partielle des normes IEEE 802 relatives à l'Internet des objets suit.

.4.1 La norme IEEE 802.11

IEEE 802.11 fait partie de l'ensemble IEEE 802 des normes techniques pour les réseaux locaux (LAN), de protocoles de contrôle d'accès au support (MAC) et de couche physique (PHY) pour la mise en œuvre d'une communication informatique de réseau local sans fil (WLAN). Cette norme définit plusieurs spécifications de couche physique (PHY) pour la connectivité sans fil pour les stations fixes, portables et mobiles (STA) dans une zone locale [5].

.4.2 La norme IEEE 802.15.15

Cette norme spécifie que la couche physique (PHY) et la sous-couche de contrôle d'accès au support (MAC) sont utilisées pour la connectivité réseau ad-hoc sans fil avec des appareils fixes, portables et mobiles à très faible consommation d'énergie. Les couches physiques (PHY) sont définies pour les dispositifs fonctionnant dans une variété de domaines règlementaires.

.4.3 La norme IEEE 802.15.4

Dans cette norme, les couches MAC et PHY sont définies de manière à répondre à des exigences telles qu'un débit de données élevé, une faible consommation d'énergie et une faible latence. Le réseau de capteurs corporels sans fil utilise le protocole d'accès multiple le plus répandu, soit le « Carrier Sense Multiple Access (CSMA/CA) », dans les réseaux sans fil avec évitement de collisions [5].

.5 Application d'Internet des objets

Aujourd'hui, l'Internet des objets est influencé par les différents scénarios d'utilisation qui sont considérés à la fois dans le monde scientifique et industriel. À titre d'illustration, quelques exemples courants sont présentés dans la figure 2.

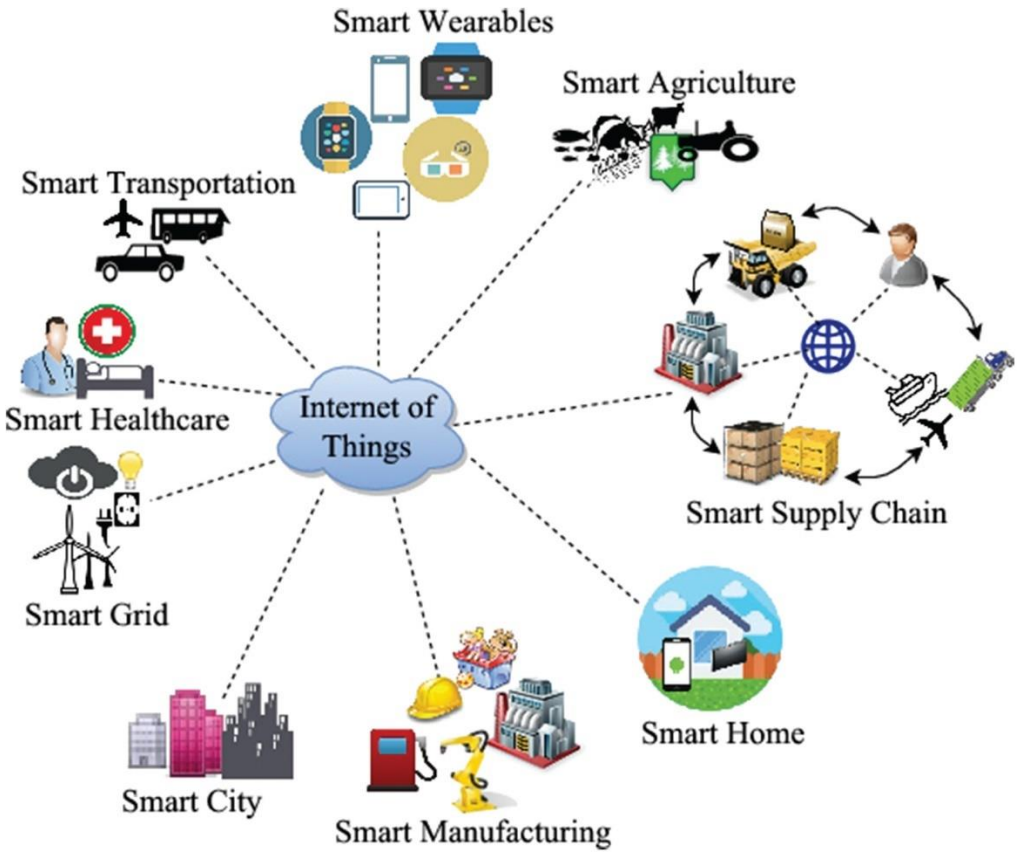


Figure 2 : Domaines d'application de l'Internet des objets [6]

Chacun d'entre eux possède ses propres objectifs et ses propres contraintes. D'où l'intérêt de considérer l'Internet des objets non pas sous le prisme d'une application spécifique, mais comme la combinaison des besoins d'une multitude de scénarios. Ces derniers peuvent prendre

place dans des espaces privés au sein desquels les interactions avec l'extérieur sont strictement contrôlées ou dans des espaces publics ouverts à tous (par exemple, la détection des places de stationnement vides), avec toute la gamme de nuances qui peut exister entre les deux. Parmi les domaines d'application d'Internet des objets, on peut citer :

.5.1 Ville intelligente (« Smart City »)

Une ville intelligente désigne une ville utilisant les technologies de l'information et de la communication pour améliorer la qualité des services urbains ou encore réduire ses coûts. Ce paradigme est capable de gérer de petits à grands volumes de données en interconnectant de minuscules capteurs, des véhicules, des appareils électroniques, des humains et des machines pour former un environnement connecté. Cette fonctionnalité est directement exploitée dans l'environnement de la ville intelligente pour activer différentes applications et assurer un support de service fiable pour les utilisateurs [7].

La sécurité dans cet environnement intelligent est une préoccupation majeure en raison de l'ouverture de la plate-forme de communication et des technologies d'interconnexion. Bien que l'interopérabilité des technologies fournisse un support de service et un accès transparent aux applications, la sécurité est un défi ouvert dans un tel environnement de ville intelligente activé par les technologies de l'information et de la communication [8].

.5.2 Réseaux intelligents (« Smart Grids »)

Les réseaux intelligents regroupent des technologies numériques et électriques qui assurent l'ajustement des flux d'électricité entre fournisseurs et consommateurs. En collectant des informations sur l'état du réseau, ce paradigme permet de contribuer à l'équilibre entre production, distribution et consommation. Les réseaux intelligents peuvent être définis selon quatre caractéristiques en termes de :

- Flexibilité : les réseaux intelligents permettent une gestion plus précise de l'équilibre entre production et consommation ;
- Fiabilité : les réseaux intelligents améliorent l'efficacité et la sécurité des réseaux ;
- Accessibilité : les réseaux intelligents facilitent l'intégration des sources d'énergie renouvelable sur l'ensemble du réseau ;
- Économie : grâce à une meilleure gestion du système, les réseaux intelligents permettent de réaliser des économies d'énergie et de réduction des coûts [9].

.5.3 Réseau de santé intelligent (« Smart Healthcare »)

L'Internet des objets (IoT) apporte une myriade d'avantages aux soins de santé et offre un lien beaucoup plus efficace entre les patients, les médecins et les produits pharmaceutiques. Le domaine médical peut aussi intégrer des applications pertinentes. L'Internet des objets peut également rationaliser les dossiers médicaux et l'accès des patients rendant les données disponibles en temps réel dans tous les services, par exemple, attacher des capteurs à un patient à domicile permet aux médecins de les surveiller à distance. L'Internet des objets a été utilisé de manière significative à de nombreux égards lors de la lutte contre la pandémie de COVID-19. La télémédecine et la surveillance de la réfrigération des vaccins étaient deux implémentations de l'IoT.

.5.4 Maison intelligente (« Smart Home »)

La domotique ou maison connectée est la technique qui permet d'automatiser certaines actions dans une maison telle que verrouiller ou déverrouiller une porte, activer ou désactiver le système de sécurité, allumer ou éteindre le chauffage ainsi que d'autres scénarios. La technologie de la maison intelligente est une tendance croissante qui permet de combiner les différentes technologies telles que la domotique et l'intelligence artificielle dans une expérience de vie quotidienne. Les technologies de réseau sans fil conduisent à la diffusion de nouvelles applications de réseau domestique pour améliorer le confort des occupants. Contrairement à la maison connectée, la maison intelligente est programmée pour agir seule. La collection et l'échange des données entre les objets connectés dans une maison intelligente sont basés sur deux principales architectures réseau :

.5.4.1 Architecture centralisée

Dans ce modèle, les appareils de l'IoT sont gérés de manière centralisée par un seul contrôleur. L'architecture centralisée permet une configuration de réseau dans laquelle les objets doivent communiquer avec un seul contrôleur central pour pouvoir communiquer entre eux. Comme tous les objets doivent passer par une seule ressource centralisée, la perte de cette ressource empêcherait tous les participants de communiquer [10]. La figure 3 présente un exemple de l'architecture centralisée.

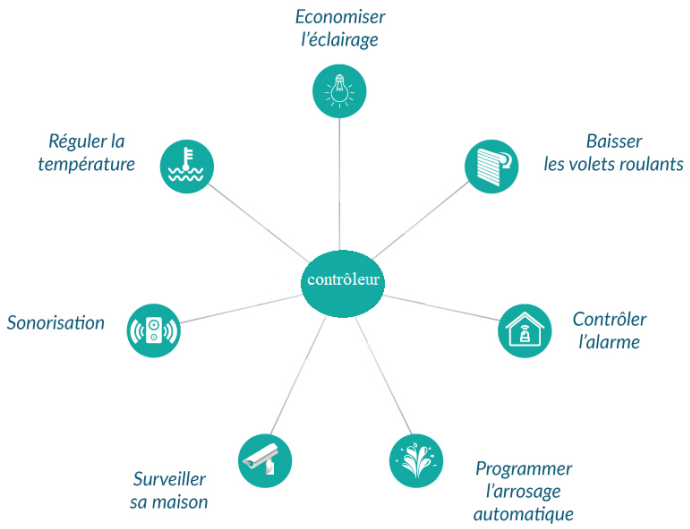


Figure 3 : Architecture centralisée

.5.4.2 Architecture décentralisée

La décentralisation fait référence au transfert du contrôle et de la prise de décision d'une entité centralisée vers un réseau distribué. Le réseau « peer to peer (P2P) » est un modèle de communication en réseau décentralisé qui se compose d'un groupe d'appareils qui stockent et partagent collectivement des données, où chaque nœud agit comme une entité individuelle. Les participants au réseau mettent à disposition une partie de leurs équipements et ressources informatiques, accessibles de manière directe par les pairs. Ces ressources partagées sont nécessaires au bon fonctionnement du service offert par le réseau. La décentralisation est l'une des fonctionnalités de la « Blockchain ». Les données de la « Blockchain » sont distribuées à chaque nœud du réseau. De ce fait, l'architecture décentralisée ne nécessite pas d'autorité centrale. Grâce à ses caractéristiques décentralisées, l'architecture décentralisée peut éviter un point de défaillance unique. De plus, l'architecture décentralisée est immuable face aux attaques, car l'attaquant doit viser chaque nœud du réseau « Blockchain »[11]. La figure 4 présente un exemple de l'architecture décentralisée.

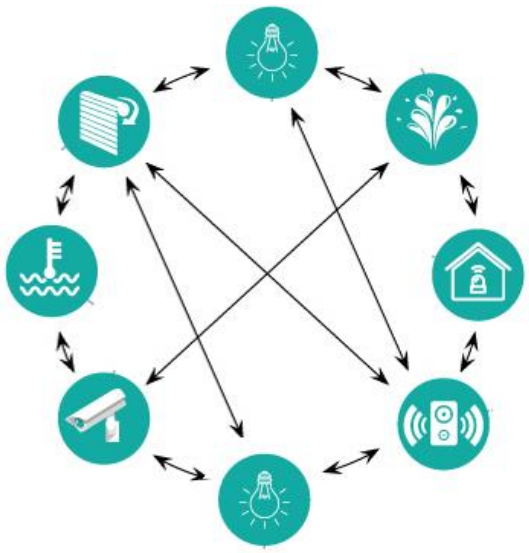


Figure 4 : Architecture décentralisée

.6 Communication Machine à Machine (M2M)

La communication machine à machine (M2M) est l'échange d'informations entre machines. Il s'agit d'une extension de l'Internet des objets (IoT) qui permet de capturer des informations sur leur état, de communiquer ces informations sur un réseau sans nécessiter d'intervention humaine et d'utiliser ces informations pour contrôler leur comportement opérationnel. Le M2M est une technologie normalisée pour gérer et partager les données ainsi que les fonctionnalités des appareils connectés. Elle est utilisée dans des applications telles que le comptage, l'automatisation, le suivi et le traçage, les soins de santé, et bien d'autres.

.7 Protocole de communication de l'IoT

La gestion d'une telle quantité de trafic nécessite de développer des protocoles de communication spécifiques capables de faire face, de manière adéquate, aux scénarios de ressources limitées du monde de l'IoT. Dans la couche d'application de l'IoT, plusieurs protocoles sont conçus spécifiquement pour les dispositifs de ressources limitées. Les deux principaux protocoles d'application conçus pour l'IoT sont « The Constrained Application Protocol (CoAP) » et le protocole « Message Queuing Telemetry Transport (MQTT) ».

.7.1 CoAP

CoAP est une version légère du protocole « Hypertext Transfer Protocol (HTTP) ». CoAP a été conçu par « Internet Engineering Task Force (IETF) » pour cibler les dispositifs de ressources contraintes à l'aide d'un sous-ensemble des méthodes HTTP, ce qui le rend fonctionnel avec le HTTP [10]. CoAP envoie les messages compressés avec le protocole « User Datagram Protocol (UDP) », supprimant toutes les exigences du protocole « Transmission Control Protocol (TCP) », ce qui réduit les exigences de la bande passante et offre plus de simplicité le rendant plus adapté aux applications de l'IoT. CoAP utilise la même architecture client-serveur que le protocole HTTP et fournit des interactions axées sur les ressources. Le protocole est conçu pour les applications de machine à machine (M2M) telles que l'énergie intelligente et l'automatisation des bâtiments[12].

.7.2 Le protocole MQTT

Le protocole MQTT est un protocole de publication et d'abonnement basé sur le protocole TCP dont l'utilisation s'est beaucoup développée ces dernières années, notamment dans le domaine des applications de l'Internet des objets (IoT) et des réseaux de capteurs sans fil (WSN). Le protocole est basé sur une entité centrale « transitaire (Broker) » qui est chargée de collecter les publications et les abonnements des clients, et de transmettre les messages entre eux. Dans le protocole MQTT, le mot « topic » fait référence à une chaîne UTF-8 que le « Broker » utilise pour filtrer les messages pour chaque client connecté. Dans un modèle de publication et d'abonnement, un client publie des informations sur un sujet (« topic ») et les autres clients peuvent s'abonner uniquement aux informations qu'ils souhaitent à travers ces sujets. Les clients du MQTT sont très petits, nécessitent un minimum de ressources et peuvent donc être utilisés sur de petits microcontrôleurs. Les en-têtes du MQTT sont petits pour optimiser la bande passante du réseau. De nombreux appareils de l'IoT se connectent sur des réseaux cellulaires peu fiables. La prise en charge des sessions persistantes par le protocole MQTT réduit le temps nécessaire pour reconnecter le client au « Broker ».

.7.3 Analyse des protocoles

Les résultats montrés dans les figures 5,6 et 7 ci-dessous sont des résultats d'évaluation des performances [13] pour le protocole MQTT et CoAP. Les résultats montrent que pour un réseau domestique constitué de plus de 15 nœuds, le protocole MQTT est considéré plus léger et efficace en termes de consommations d'énergie, de cycle d'utilisation et de paquets reçus par rapport à CoAP.

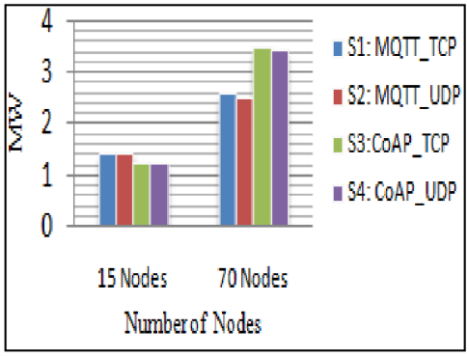


Figure 5 : Consommation d'énergie

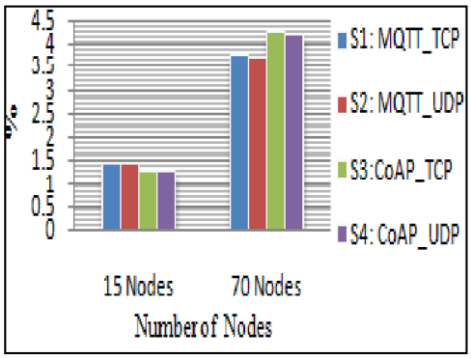


Figure 6 : Cycle d'utilisation

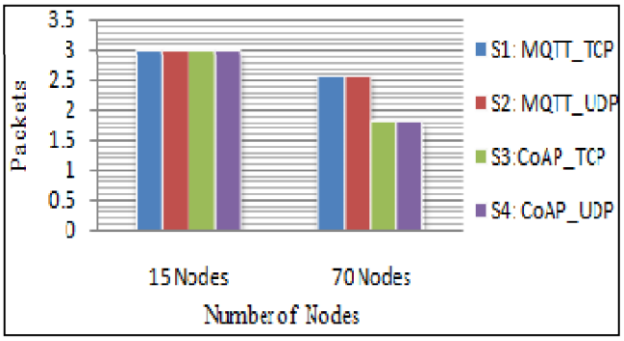


Figure 7 : Paquets reçus

.8 La sécurité des objets connectés à la maison.

.8.1 Exigences de la sécurité dans les réseaux de l'IoT

Nous discuterons dans cette section des principales exigences et des autres défis qui se posent à la sécurité des réseaux de l'IoT à la maison.

.8.1.1 Confidentialité

Ce concept de sécurité garantit que les données envoyées au sein du réseau domestique ne sont pas transmises à des parties non autorisées. Seules les parties autorisées peuvent y accéder sur le réseau. Ceci peut être réalisé en cryptant les informations à l'aide de divers algorithmes de cryptage. Cela empêche ainsi les objets malveillants de suivre et d'écouter les messages concernant les objets connectés dans le réseau domestique.

.8.1.2 Authentification

L'authentification permet aux entités du réseau de l'IoT de garantir la bonne identité des entités avec lesquelles elles communiquent. L'authentification permet aux diverses parties du réseau de l'IoT de faire confiance à la distribution des données et des messages. Il existe deux types d'authentification : l'authentification des messages qui permet de suivre l'origine du message et l'authentification d'entités qui permet d'identifier les objets du réseau domestique.

.8.1.3 Intégrité

L'intégrité des données désigne également la sureté des données. Ce concept de sécurité garantit que les informations transmises au sein du réseau de l'IoT ne peuvent pas être volontairement ou accidentellement altérées par des entités non autorisées entre la phase d'émission et de réception. L'intégrité peut être réalisée grâce à l'utilisation de la cryptographie, des fonctions de hachage ainsi que des signatures numériques.

.8.1.4 La non-répudiation

Ce concept de sécurité garantit avec certitude qu'un message a été envoyé par le véritable expéditeur et reçu par l'entité destinateur. La non-répudiation permet au destinataire d'être certain que l'expéditeur est l'auteur des messages qu'il a générés. Cela élimine la possibilité qu'un attaquant injecte des données erronées sans être identifié immédiatement. Pour atteindre cet objectif, la cryptographie et les signatures numériques sont utilisées dans l'IoT afin d'assurer la non-répudiation des messages concernant les applications de sécurité des réseaux domestique. Malheureusement, ce concept de sécurité n'est pas pris en considération dans les communications M2M.

.8.1.5 L'anonymat

L'anonymat dans l'IoT est la technique qui garantit l'identification des machines qui ont reçu ou transmis de l'information avec les autres entités. En fait, l'anonymat permet d'assurer que la vraie identité d'un objet connecté reste inconnue. Révéler l'identité des objets connectés entrainent des conséquences catastrophiques pour la sécurité de l'IoT. Cela permet à un attaquant de tracer toutes les informations de l'entité émettrice. Malheureusement, l'anonymat n'est pas considéré dans les systèmes M2M.

.8.2 Types d'attaques associées aux systèmes de l'IoT

Dans cette section, nous passons en revue les attaques malveillantes les plus courantes qui ciblent l'écosystème de la maison intelligente et décrivons certaines des limitations ainsi que certains des obstacles qui peuvent survenir. Nous discutons des moyens de les minimiser autant que possible.

Quelques-unes des principales attaques susceptibles de cibler n'importe quel système de maison intelligente sont présentées dans ce qui suit.

- **Menaces du système interne** : comprend les défaillances du système domestique, des appareils ou des pannes d'électricité et de la consommation d'énergie [14].
- **Déni-de-Service « Denial-of-Service (DoS) »** : une des menaces de sécurité qui cible la disponibilité du système en transmettant des requêtes excessives dans un court laps de temps [15, 16].

- **Homme du milieu « MAN-in-the-middle (MiM) » :** considère une attaque par usurpation qui vise à voler les demandes d'authentification d'un appareil reconnu.
- **Injection de codes malveillants :** peut être utilisé pour injecter dans la maison intelligente des scripts malveillants qui permettent aux attaquants d'utiliser la vulnérabilité du système et d'autoriser l'accès à des éléments non approuvés dans le système.

Dans ce travail, nous nous concentrons sur les attaques MiM et l'injection de codes malveillants qui sont considérés comme deux des principales menaces de sécurité visant les objets connectés.

.9 Mécanisme de la sécurité informatique

Après avoir discuté des exigences de sécurité dans les réseaux de l'IoT, cette section aborde les mécanismes appropriés et les techniques cryptographiques existantes qui peuvent fournir des solutions aux problèmes liés à l'authentification, à l'intégrité et à la non-répudiation.

.9.1 Cryptographie

La cryptographie est le domaine qui décrit les techniques utilisées pour chiffrer les données, souvent à l'aide d'une clé privée afin d'assurer la confidentialité des informations échangées. La cryptographie consiste à chiffrer le contenu des messages échangés à l'aide des algorithmes de cryptographie. Pour déchiffrer ces messages, l'entité destinatrice utilise des algorithmes de déchiffrement pour reconstruire le message d'origine. Deux types de cryptographie existent : la cryptographie symétrique et la cryptographie asymétrique.

.9.1.1 Cryptographie symétrique :

La cryptographie symétrique est considérée comme la plus ancienne technique de cryptographique. Ce type de cryptographie utilise une clé unique pour le processus de chiffrement et de déchiffrement des messages.

.9.1.2 Cryptographie asymétrique :

La cryptographie asymétrique est un type de cryptographie qui utilise deux clés de chiffrement. Une clé publique partagée avec tout le monde utilisé pour crypter les informations envoyées et une clé privée utilisée pour décrypter ces informations.

.9.2 Cryptographie à courbe elliptique (ECC)

La cryptographie ECC est une famille moderne de cryptosystèmes à clé publique qui est basée sur les structures algébriques des courbes elliptiques. La ECC implémente toutes les fonctionnalités majeures des cryptosystèmes asymétriques : cryptage, signatures et échange de clés.

La ECC utilise des clés et des signatures plus petites que le chiffrement Rivest-Shamir-Adleman (RSA) pour le même niveau de sécurité et fournit une génération ainsi qu'un accord de clé et des signatures rapidement [17].

.9.2.1 Le point générateur dans la ECC

Pour les courbes elliptiques sur des corps finis, les cryptosystèmes ECC définissent un point prédéfini (constant) spécial, appelé point générateur G (point de base), qui peut générer n'importe quel autre point de son sous-groupe sur la courbe elliptique (illustrée à la figure 8) en multipliant G par un nombre entier allant de [0... n].

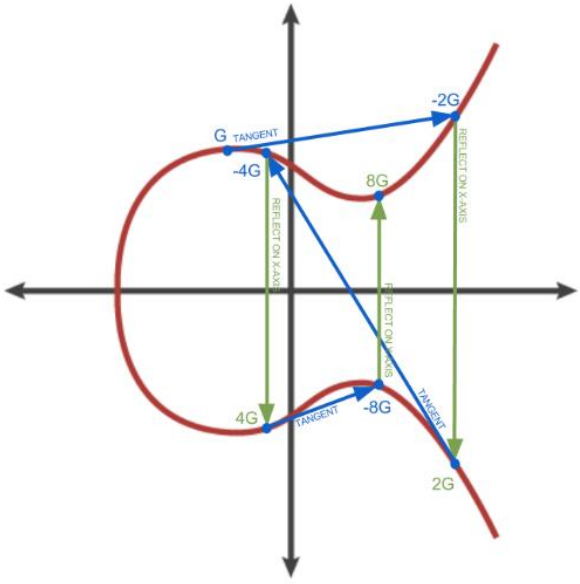


Figure 8 : Le point générateur dans la ECC [18]

.9.2.2 Échange de clés sur la courbe elliptique de Diffie-Hellman (ECDH)

L'ECDH est un schéma d'accord de clé anonyme qui permet à deux parties, qui ont chacune une paire de clés publique-privée à courbe elliptique, d'établir un secret partagé sur un canal non sécurisé [19]. L'ECDH est très similaire à l'algorithme classique de l'échange de clés de Diffie-Hellman (DHKE), mais il utilise la multiplication de points dans la ECC au lieu d'exponentiations modulaires.

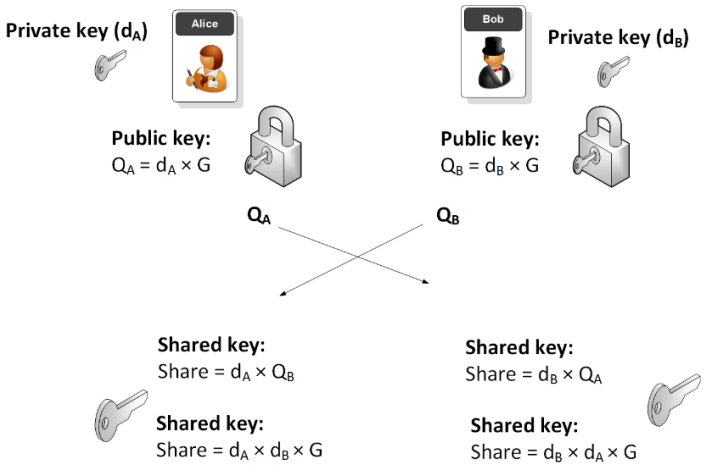


Figure 9 : Elliptic Curve Diffie–Hellman Key Exchange [20]

.9.3 Fonctions de hachage

Les fonctions de hachage permettent de calculer, à partir d'une chaine de données fournie en entrée, une empreinte digitale afin d'identifier rapidement la donnée initiale. L'entrée de la fonction de hachage est de longueur arbitraire, mais la sortie est toujours de longueur fixe. Les fonctions de hachage les plus utilisées sont celles à sens unique et les valeurs renvoyées par ces fonctions sont appelées valeurs de hachage.

.9.4 Signature numérique

Une signature numérique est une donnée associée à un message qui permet de garantir l'authenticité, l'intégrité et la non-répudiation du message. Celle-ci s'appuie sur la cryptographie asymétrique. En effet, la signature numérique est générée par une fonction de hachage et une clé privée de l'expéditeur, tandis que le destinataire vérifiera l'intégrité et l'authenticité du message en utilisant la clé publique correspondante à l'expéditeur.

.9.5 Certificats numériques

Le certificat numérique dans l'IoT est une carte d'identité numérique générée par la fonction de hachage et la clé privée de l'émetteur dont le but est d'identifier des objets physiques ou non-physiques. Un certificat numérique est délivré par un tiers de confiance appelé l'autorité de certification (AC). Ce dernier permet de renforcer la sécurité dans les réseaux de l'IoT en attestant l'authenticité des paires de clés publiques et privées ainsi qu'en identifiant les objets connectés d'une façon unique. Le certificat numérique contient les informations suivantes :

- Numéro de série du certificat ;
- Nom de l'entité qui possède la clé publique ;
- Clé publique liée à une entité ;
- Nom de l'autorité de certification qui l'a publiée ;
- Durée de validité de certificat ;
- Algorithme de cryptage ;
- Restrictions d'utilisation de la clé publique.

.10 Conclusion

La sécurité des informations a une grande importance dans tous les réseaux, notamment dans les réseaux de l'IoT. Dans ce réseau, il y a plusieurs informations échangées entre les objets connectés sans interférence humaine telles que les systèmes M2M. Ces données sont sujettes aux risques des cyberattaques. Dans ce chapitre, nous avons abordé les caractéristiques et les applications de l'IoT ainsi que les différents types d'attaques et de mécanismes de sécurité. Dans le chapitre suivant, nous allons aborder un certain nombre de recherches liées à la sécurité de l'IoT ainsi qu'aux systèmes M2M en discutant des points forts et des points faibles de chaque modèle proposé.

CHAPITRE 3 REVUE DE LA LITTÉRATURE

L'Internet des objets (IoT) est l'une des technologies fondamentales sur lesquelles repose la quatrième révolution industrielle [21]. Le concept de l'IoT a évolué d'une simple technologie favorable à un besoin technologique majeur dans la vie humaine quotidienne. Dans la littérature de l'IoT, il y a eu une augmentation des projets de recherche axés sur la maison intelligente. Les projets de maison intelligente sont conçus pour aider à automatiser les tâches humaines quotidiennes en réduisant l'effort manuel [22].

Dans cette partie, nous présentons un certain nombre de recherches issues de la littérature liées à la sécurité de l'IoT dans une maison intelligente. Nous discutons des défis associés aux architectures réseau, de leurs protocoles ainsi que de leurs communications M2M afin de garantir la sécurité des communications dans une maison intelligente.

.1 IoT (architecture de la maison intelligente)

Dorri, Kanhere et Jurdak (2016) ont proposé une architecture hiérarchique qui se compose de maisons intelligentes, d'un réseau superposé et de stockages en nuage coordonnant les transactions de données avec la « Blockchain » pour assurer la confidentialité et la sécurité des données. Dans ce modèle, les appareils de l'IoT étaient gérés de manière centralisée par un seul contrôleur. Cette architecture assure la confidentialité des informations échangées [19], mais malheureusement leur modèle ne prend pas en compte la non-répudiation et la sécurité contre certaines vulnérabilités telles que les attaques de l'homme au milieu (MiM). De plus, le modèle centralisé souffre de la surcharge de calcul sur le mineur central, ce qui est l'un des aspects négatifs de l'énergie dans l'IoT.

Gallo et al. ont discuté sur un scénario d'interconnexion de réseaux typiques basé sur un nœud central, appelé « Home Gateway (HGW) », et des solutions basées sur le protocole Internet (IP) classique qui utilise l'authentification comme technique de sécurité la plus répandue [23]. La gestion de la confiance utilisée dans cet article gagne en popularité en raison de sa capacité à empêcher ou à détecter les nœuds malveillants [23].

Par contre, il y a plusieurs problèmes liés à ces solutions. En effet, le mécanisme de contrôle d'accès, le temps nécessaire pour récupérer les données de différents appareils ainsi que les protocoles utilisés ne peuvent pas être appliqués aux applications futures, car ils sont basés sur une architecture centralisée. Cette architecture centralisée peut avoir un point de défaillance unique avec la surcharge de calcul.

Han, Kim et Jang ont proposé un système de verrouillage de porte intelligent (SLD) basé sur la « Blockchain » pour assurer l'intégrité des données à l'aide d'un algorithme afin que le système SLD évalue certaines situations dans son environnement et fonctionne en se basant sur les données envoyées par les capteurs [24]. L'algorithme proposé assure l'intégrité des données, mais il ne prend pas en compte la non-répudiation et l'anonymat. Cet algorithme est conçu uniquement pour un petit réseau et il manque de robustesse.

DeCyMo est une architecture proposée par Gallo et al. qui vise à résoudre les problèmes et les vulnérabilités courants de l'IoT en tirant parti de la structure centralisée de l'IoT et de l'architecture distribuée de la « Blockchain » pour l'application de méthodologies de réseau « Software-Defined Networking (SDN) » et pour les applications distribuées sécurisées [25]. Le schéma proposé garantit l'intégrité et la confidentialité des données. Malheureusement, ce dernier n'est applicable que sur des machines à forte puissance de calcul. En effet, dans un réseau domestique, plusieurs objets connectés ne répondent pas à ces exigences à cause de leur faible stockage énergétique. Les capteurs de l'IoT ne peuvent pas facilement stocker une « Blockchain », ce qui rend les objets connectés vulnérables aux attaques qui visent à voler les données. De plus, l'architecture proposée n'assure pas la non-répudiation pour les communications M2M.

Aung et Tantidham ont présenté une approche de mise en œuvre d'une « Blockchain » privée pour un système de maison intelligente afin de faire face à ses problèmes de confidentialité et de sécurité [26]. Dans le système proposé, un contrat intelligent est utilisé pour maintenir les transactions de données. Donc, l'exploration n'est pas nécessaire. Par conséquent, le système proposé n'a pas besoin d'utiliser un ordinateur de grande puissance de traitement. Les auteurs ont discuté des politiques de contrat

intelligent pour le système de maison intelligente avec un temps de transaction de 20 secondes [26], ce qui ne convient pas aux conditions de l'environnement des appareils qui fonctionnent en temps réel.

Hendawi et al. ont proposé un correctif de sécurité matérielle et logiciel (HW/SW) en créant un prototype « boutons intelligents » qui combine des capteurs, des actionneurs, des processeurs et de la mémoire à un logiciel pour protéger les appareils de l'IoT contre une ou plusieurs attaques de sécurité [27]. Le bouton intelligent se compose essentiellement d'un « Raspberry Pi » et d'un ensemble de capteurs connectés [27]. Le correctif proposé est efficace contre quelques types d'attaques associées aux caméras de l'IoT et aux robots. Malheureusement, ce dernier est coûteux en termes de consommation d'énergie, car le bouton intelligent est composé de plusieurs dispositifs électroniques qui nécessitent plus de ressources énergétiques.

Irman et al. ont proposé un système extensible de portail automatique domestique intelligent sans fil utilisant l'IoT. Ce système utilise la combinaison d'une application mobile et d'un serveur « cloud » [28]. L'architecture du système comprend à la fois des logiciels et du matériel utilisés pour permettre de transformer une porte automatique standard en une porte automatisée intelligente pour l'accès quotidien des utilisateurs à domicile. Le contrôle à distance s'effectue à l'aide d'une application mobile basée sur « Android ». L'architecture du système permet donc la création d'un système hybride.

Al-Syouf, Al-Duwairi et Shatnawi ont proposé un système qui aidera à surmonter un certain nombre d'attaques dirigées contre les réseaux domestiques intelligents [29]. Le système proposé constitue un environnement de maison intelligente, de calcul de brouillard et de réseaux « cloud » à l'aide de certains protocoles à la couche de base pour prendre en charge la redondance en cas de basculement. Le protocole utilisé est basé sur une technique de « tokenization » des requêtes entre le client et le serveur Web virtuel pour détecter des attaques par injection « Structured Query Language (SQL) » dirigée vers le serveur Web. Le système de sécurité utilisé dans l'infonuagique est basé sur le hachage entre le serveur Web virtuel et le serveur principal pour empêcher l'accès aux données sensibles. Les

résultats expérimentaux montrent que le schéma proposé réduit le nombre d'attaques qui peuvent cibler les utilisateurs pendant qu'ils naviguent sur le Web [29]. Cependant, ce dernier ne prend pas en compte les risques d'attaques par des objets connectés malveillants (attaques MiM) qui peuvent changer l'état d'une machine dans une maison intelligente sans l'accès par navigation Web.

.2 IoT (cryptographie)

Kothmayr et al. ont proposé un schéma de sécurité d'authentification bidirectionnelle pour l'Internet des objets (IoT) basé sur les normes Internet existantes, en particulier le protocole « Datagram Transport Layer Security (DTLS) » et le cryptage Rivest-Shamir-Adleman (RSA) [30]. Le schéma est conçu pour être utilisé sur le protocole Internet version 6 (Ipv6) et sur les réseaux personnels sans fil à faible puissance (6LoWPAN). L'architecture proposée offre l'intégrité, la confidentialité et l'authenticité des messages avec une énergie abordable, une latence de bout en bout ainsi qu'une surcharge de mémoire. Malheureusement, le schéma proposé ne prend pas en considération la non-répudiation et l'anonymat. De plus, les auteurs ont utilisé l'algorithme de chiffrement RSA [30]. Ce dernier est plus coûteux en termes de calcul par rapport aux autres algorithmes de cryptage. Relativement plus récente que le RSA et l'« Advenced Encryption Standard (AES) », la cryptographie à courbe elliptique (ECC) a cessé de gagner du terrain en tant que solution qui peut concilier l'efficacité des ressources de l'AES à la robustesse du RSA pour les appareils à ressources limitées.

Wang et al. ont proposé une étude des systèmes de cryptage basé sur les attributs « attribute-based encryption (ABE) ». Selon les auteurs, un schéma de cryptage ABE se compose d'une autorité de clé, d'éditeurs (expéditeurs) et d'abonnés (destinataires) [31]. L'autorité de clé authentifie les éditeurs et les abonnés (vérifie qu'ils sont bien qui ils prétendent être ainsi que leurs attributs), génère des clés publiques ainsi que des clés privées, et délivre les clés aux éditeurs et aux abonnés. Pour tester l'efficacité de ce type de systèmes dans un environnement de l'IoT, les auteurs ont implémenté le système sur différents types de machines et ont étudié son efficacité en termes de temps d'exécution et de consommation des ressources [31]. Les résultats d'analyse ont montré que ce type de

système est coûteux en termes de calcul en raison de leur complexité et nécessitent de nombreux cycles de chiffrement. Cependant, le système est raisonnable pour être exécuté dans un environnement « ordinateur » puissant. Malheureusement, les auteurs n'ont pas étudié l'efficacité de ce type système face aux différents types d'attaques d'écoute.

Hussain a proposé une méthode pour renforcer la sécurité du schéma RSA. L'objectif de cette méthode proposée est d'éliminer les messages redondants qui se produisaient pour certaines valeurs de n (où n est nombre entier) [32]. En effet, lorsque n correspond au produit de deux nombres premiers, les messages redondants se produisent. Alors, la méthode proposée remplace d'abord cette valeur de n en utilisant une distance d'accord sécurisée dans un ensemble de tous les nombres premiers disponibles. Ensuite, la méthode consiste à sélectionner un des nombres premiers responsables de la génération d'un n alternatif ou à sélectionner les deux nombres premiers et faire la génération à partir de cet ensemble. Cette méthode élimine également la valeur de n de la clé publique [32]. Dans leur solution, la clé publique est partagée avec tout le monde tandis que la clé privée est gardée secrète ce qui empêche les attaques multiples. Toutefois, le chiffrement et le déchiffrement sont plus lents, ce qui ne convient pas aux conditions de l'environnement des appareils qui fonctionnent en temps réel. De plus, le niveau de sécurité obtenu n'est pas satisfaisant.

Koppula et Muthukuru ont conçu un mécanisme d'authentification pour sécuriser la signature numérique basée sur les courbes elliptiques pour l'IoT [33]. L'avantage de l'utilisation de la cryptographie à courbe elliptique est que la taille de la clé est considérablement réduite par rapport aux crypto-systèmes traditionnels (tels que le RSA), ce qui améliore la sécurité du réseau [33]. La signature numérique joue un rôle important dans l'atteinte de l'intégrité, de la non-répudiation et de l'authentification des données transmises sur le réseau. Le schéma proposé est mal implémenté dans la génération de signature, car ce schéma a été développé sans inversion modulaire dans les algorithmes de génération et de vérification de signature, ce qui prend du temps pour les appareils aux capacités limitées.

Ma, Wu et Zhou ont fait une étude sur l'applicabilité des mécanismes existants de la cryptographie à clé publique et à clé pré-partagée ainsi que sur l'applicabilité de ces systèmes de gestion de clés afin de générer des clés partagées pour les nœuds de capteurs (WSN) dans le contexte de l'IoT [34]. Les auteurs utilisent un schéma de signature et de cryptage pour proposer un modèle de transmission qui parvient à répondre aux exigences de sécurité pour l'anonymat et la confidentialité dans l'IoT [34]. Dans ce modèle, la non-répudiation n'est pas assurée, car le modèle est vulnérable contre certains types d'attaques sur le réseau qui visent à voler les informations pendant la transmission des données. Ce type d'attaque peut avoir de graves effets sur le système.

.3 M2M

Ben Saied et al. ont proposé un schéma qui porte sur l'établissement d'une clé symétrique entre un dispositif M2M contraint et un serveur distant [35]. Un nœud M2M qui ne peut pas gérer des opérations asymétriques lourdes confie celles-ci à des voisins moins contraints, appelés proxys, pour qu'ils les fassent en son nom. En effet, ces mandataires assurent la livraison sécurisée d'un secret aléatoire du nœud contraint au serveur distant et inversement. Sur la base de ces deux secrets échangés, les deux entités calculent la clé symétrique utilisée pour sécuriser les communications ultérieures. Pour empêcher que le secret aléatoire d'un nœud contraint ne soit connu, il est divisé en différents partis et une seule partie est fournie à chaque proxy chiffrée avec la clé pré-partagée. Une fois reçue, chaque proxy utilise sa paire de clés éphémères (une publique et une privée) qui est fournie par un Tiers de Confiance (TTP) [35]. Le schéma proposé souffre de deux problèmes principaux. Tout d'abord, ils nécessitent que le nœud M2M contraint emmagasine et gère un grand nombre de clés. Deuxièmement, le schéma repose sur une hypothèse qui consiste à faire confiance aux proxys. Si les proxys collaborent, ils seraient capables de calculer la clé symétrique précédemment établie.

L'approche de gestion des clés de Hussan et al. est spécialement conçue pour les appareils utilisant le protocole Internet version 6 (IPv6) sur les réseaux personnels sans fil à faible consommation (6LoWPAN) [36]. Dans cette solution, les différents équipements sont classés en trois catégories selon leurs ressources et leur rôle : « End Device (6ED) » les plus contraints en ressources, « Router (6LR) » les moins contraints et « Border Router

(6LBR) » pour gérer l'authentification. Leur schéma se compose de deux phases : une phase d'authentification et une phase de génération de clé. Dans la première phase, le 6LBR authentifie à la fois les nœuds 6LR et 6ED. Une fois authentifié, le 6LBR fournit en toute sécurité aux 6LR une paire de clés publique et secrète de chiffrement à courbe elliptique (ECC) qui sera utilisée lors de la phase suivante. La Deuxième phase consiste à établir une clé symétrique qui sécurisera les communications ultérieures entre le 6ED et le serveur distant. Cette clé est générée à l'aide de l'accord d'échange de clés sur la courbe elliptique de Diffie-Hellman (ECDH) [36]. En effet, dans le schéma proposé, la non-répudiation n'est pas assurée, car une fois que la génération de clé est effectuée, le 6ED et le serveur distant entrent dans une communication chiffrée avec une seule clé. De plus, le schéma repose sur une hypothèse qui consiste à faire confiance aux 6LR et 6LBR, alors que ces derniers peuvent déchiffrer tous les messages échangés entre le 6ED et le serveur à l'aide de la clé secrète établie. Par conséquent, le schéma proposé est vulnérable aux différents types d'attaques tels que les attaques sur 6LR et 6LBR comme l'usurpation d'identité et les attaques d'écoute.

Chae et al. ont proposé un schéma de gestion des clés pour la communication cellulaire M2M qui peuvent être connectée à la même station de base (eNodeB) ou à des eNodeB différents attachés à la même passerelle de service (SGW) [37]. La communication cellulaire M2M connectée à la même station de base génère une paire de clés et la fournit en toute sécurité aux deux nœuds chiffrés afin de permettre la sécurisation des communications mobiles. Dans le cas de la communication cellulaire M2M connectée à des eNodeB différents qui sont attachés à la même passerelle de service (SGW), la clé est générée par la passerelle de service et envoyée aux deux eNodeB qui la transmettent aux appareils M2M. Lorsque les appareils sont connectés à différents eNodeB, les appareils M2M commencent à échanger des paquets chiffrés avec une clé temporaire. Une fois qu'ils ont reçu la clé partagée, les clés temporaires sont échangées et chiffrées avec la clé partagée [37]. Un tel schéma n'est pas toujours évolutif, car la mise à jour de la clé de groupe nécessite l'échange de plusieurs paquets. Donc, leurs solutions ne peuvent être appliquées que lorsqu'il s'agit d'appareils détenant une carte SIM et des clés associées, car il n'y a

aucun moyen de sécuriser l'échange de clé et d'empêcher les indiscrets de connaitre la clé. Cette exigence n'est pas satisfaite par la plupart des appareils M2M.

Barthe et al. ont proposé un schéma d'authentification capable de rejeter les paquets non intentionnels ou non légitimes juste après la réception du préambule physique. Le préambule d'authentification (AP) annexé se compose des 32 premiers bits de la sortie d'un code d'authentification de message basé sur le hachage (HMAC) [38]. Le premier AP est généré grâce à une fonction HMAC ayant en entrée les identités des deux nœuds communicants en combinaison avec une clé partagée par paire. Le protocole de synchronisation proposé résout les problèmes de désynchronisation précédents et ajoute les paramètres de déploiement qui maximisent les économies d'énergie globales en ce qui concerne, à la fois, les attaques par épuisement des nœuds et le fonctionnement normal du réseau. Les auteurs ont également parlé des processus de génération et de mise à jour des clés nécessaires pour gérer le matériel de chiffrement utilisé [38]. De plus, les auteurs ont mentionné que ces processus montrent comment intégrer le mécanisme proposé dans l'amendement IEEE 802.15.4e à la norme IEEE 802.15.4-2006 [38]. De nombreuses entreprises ont décidé d'opter pour cette technologie pour le développement de réseaux M2M. Cependant, cette proposition souffre d'un défaut majeur, car l'AP ajouté ne dépend pas du contenu du paquet. Ainsi, un adversaire peut supprimer des paquets légitimes et ajouter son point d'accès à n'importe quel message de son choix. De plus, cela nécessite la distribution de la même clé à tous les nœuds avant leur déploiement. Une telle solution ne peut être utilisée que lorsqu'il s'agit de nœuds stationnaires dont le voisinage ne change pas.

Une analyse de diverses architectures de réseaux domestiques dans la littérature de l’IoT peut identifier les avantages et les lacunes de ces architectures. Aussi, l’analyse des techniques de sécurité proposées dans la littérature M2M révèle de nouvelles attaques et vulnérabilités qui pourraient poser des risques importants dans les maisons intelligentes.

À notre connaissance, il n'existe aucun protocole dans la littérature de l’IoT qui puisse satisfaire toutes les exigences de sécurité informatique, y compris la non-répudiation. Aussi, aucune architecture domestique ne prend en considération l’optimisation du coût de calcul pour

les objets à faible capacité de calcul. Dans le chapitre suivant, nous allons présenter notre modèle suivi par les différents scénarios d’attaques.

CHAPITRE 4 MODÉLISATION ET SCÉNARIOS D'ATTAQUES

Dans cette partie, nous présentons notre architecture ainsi que plusieurs scénarios d'attaques qui peuvent être lancés via des appareils de l'IoT. Nous présentons aussi notre protocole de chiffrement afin de voir l'impact de ces attaques sur notre modèle à travers une vérification formelle à l'aide de l'outil « ProVerif ».

.1 Architecture proposée

La structure globale de notre modèle (Figure 10) se compose de deux couches principales, externe et interne, liées entre eux avec la passerelle IoT (IoT gateway).

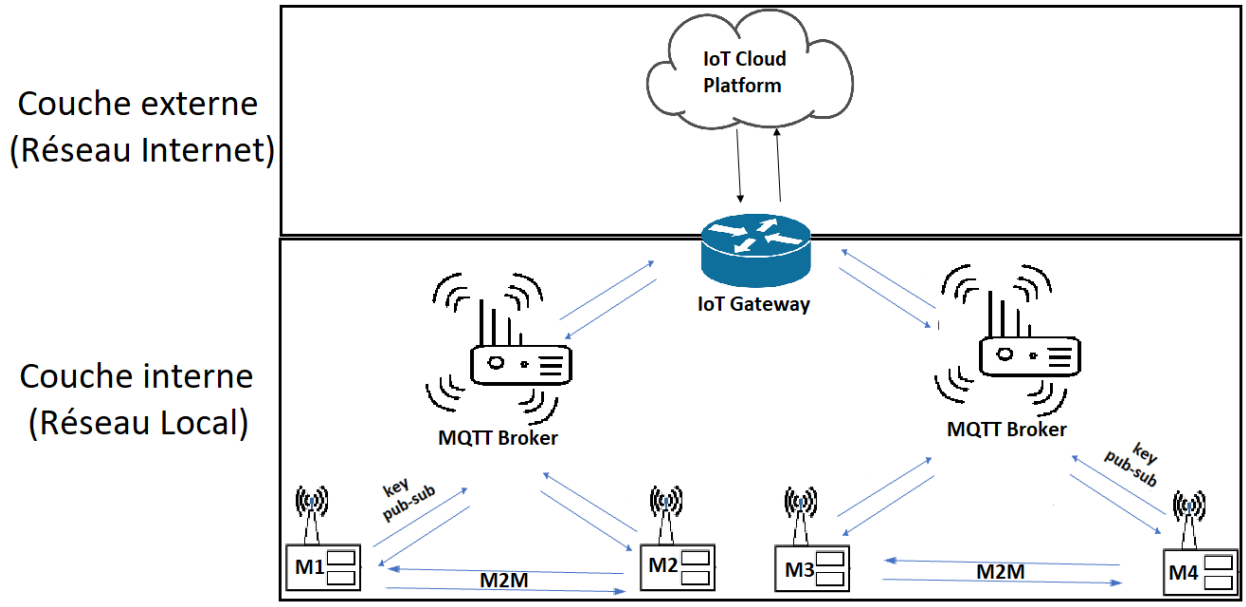


Figure 10 : Architecture globale du système

La couche externe représente les plate-formes « Clouds » de l'IoT et tous les services qui peuvent être affectés afin de transférer et collecter des données de la couche interne de la maison vers les utilisateurs distants pour assurer divers scénarios de notification et de contrôle.

La couche interne est un réseau local composé de plusieurs « Brokers » MQTT afin de réduire le coût de calcul sur un seul contrôleur central. De plus, la distribution des données entre « Brokers » MQTT améliore également l'intégrité des données au sein du réseau domestique. Chaque « Broker » MQTT est connecté à plusieurs machines afin de gérer les machines d'une manière centralisée pour assurer la mise à jour des clés de chiffrement, l'authentification d'appareils et le partage des données sur l'« IoT Cloud platform ». Cette couche du réseau fait également partie de l'architecture « Peer to Peer (P2P) » pour les communications M2M afin de renforcer la confiance, la non-répudiation et l'anonymat entre les pairs (M1, M2, M3 et M4 voir la figure 10).

La génération et l'échange des clés de chiffrement pour les communications M2M sont effectués par les machines elles-mêmes. La mise à jour des clés de chiffrement pour les machines non participantes à une telle communication se fait par les « Brokers » MQTT selon le modèle de publication et d'abonnement du protocole MQTT, où chaque « Broker » MQTT contient un nombre de sujets (« topics ») qui varie selon le nombre de machines connectées autour de lui. Chaque sujet est associé à une machine spécifique pour assurer l'authentification entre le « Broker » MQTT et les machines.

Après chaque transaction valide entre deux machines, les machines participantes à une communication M2M doivent notifier le « Broker » MQTT de la mise à jour de ses clés de chiffrement. Le « Broker » MQTT fait un filtrage de toutes les notifications entrantes et les distribue correctement aux machines abonnées selon le modèle de publication et d'abonnement afin que les machines fassent la mise à jour des clés. Une machine est abonnée à un ou plusieurs sujets afin qu'elle reçoive correctement la notification de mise à jour des clés.

Pour surpasser les défis liés à la non-répudiation des données, à l'anonymat et à l'authentification des communications M2M, notre algorithme de chiffrement est basé sur les courbes elliptiques ainsi que sur l'utilisation de l'accord d'échange des clés de Diffie-Hellman (ECDH) afin d'utiliser des clés plus petites pour réduire le coût de calcul pour les machines à faible puissance de calcul. Ce dernier est renforcé par les signatures numériques de la cryptographie à courbes elliptiques (ECC), ce qui offre un niveau amélioré d'authentification et de confidentialité.

Les données seront chiffrées et déchiffrées d'une manière dynamique, où chaque machine qui participe à une communication M2M génère une nouvelle clé de chiffrement après chaque transaction valide entre deux machines. Notre système renforce le secret de clé de chiffrement ainsi que la sécurité des données et élimine les attaques d'écoute. L'algorithme de génération parallèle de clé de chiffrement entre les deux machines est le suivant :

```
Algorithme MACHINE 2

VAR
Int d : privé
Int G : public
Begin
 P= d*G;
 out(P);
 In(Q);
 R= P*Q;
 While(n<rangKey)
        Rn = (n+1)²R -2(n)R
        out(mEnc(msg,Rn));
        in(msg);
        Message=mDec(msg,Rn);
        n++;
  end
end
```

```
Algorithme MACHINE 1

VAR
Int e : privé
Int G : public
Begin
 Q= e*G;
 out(Q);
 In(P);
 R0= P*Q;
 While(n<KeyLength)
        Rn = (n+1)²R -2(n)R;
        in(msg);
        Message=mDec(msg,Rn);
        out(mEnc(msg,Rn))
       n++;
  end
end
```

La figure 11 montre le mécanisme d'échange de clés entre deux machines selon l'algorithme ECDH et la génération de clés. Les étapes sont les suivantes :

1. M1 génère une paire de clés ECC aléatoires : e (Privé), $Q = e \cdot G$ (Public)
2. M2 génère une paire de clés ECC aléatoires : d (Privé), $P = d \cdot G$ (Public)
3. M1 et M2 échangent leurs clés publiques via le canal non sécurisé
4. M1 calcule la clé partagée $R = e \cdot P \implies e \cdot (d \cdot G)$
5. M2 calcule la clé partagée $R = d \cdot Q \implies d \cdot (e \cdot G)$
6. Maintenant, M1 et M2 ont la même clé partagée (privé) $R = e \cdot P = d \cdot Q$
7. M1 envoie un message signé et chiffré avec R à M2
8. M2 déchiffre le message envoyé par M1 et vérifie sa signature afin de l'accepter ou non
9. Après chaque transaction n valide, les machines M1 et M2 génèrent une nouvelle clé de cryptage et de décryptage $R_n = (n+1)^2R - 2(n)R$, et envoient une requête au « Broker » MQTT pour qu'il assure lui-même la mise à jour des clés avec les autres machines connectées

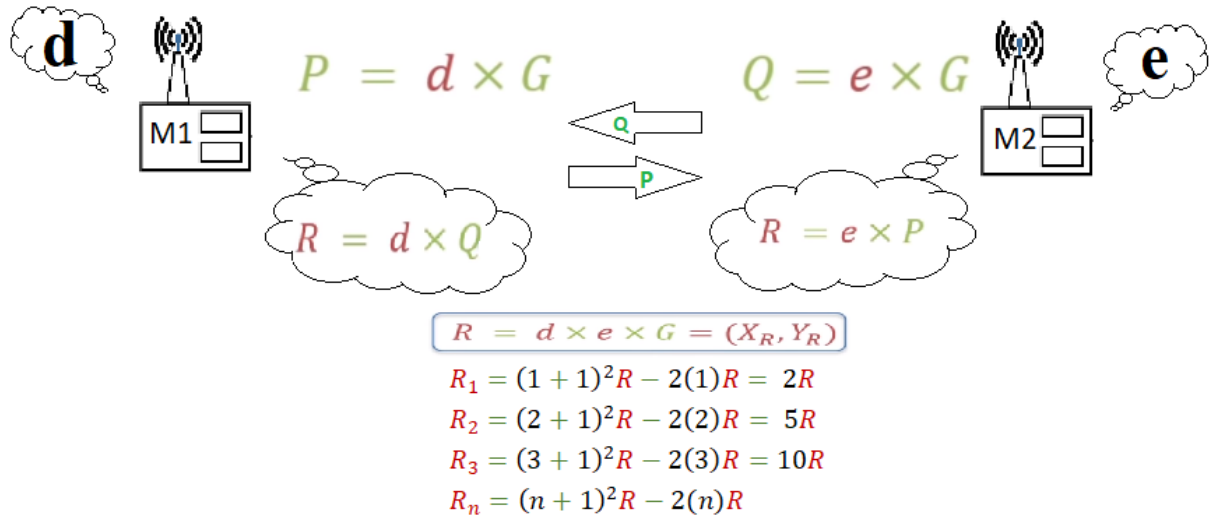


Figure 11 : Algorithme ECDH avec la Génération de clés

Tous les messages envoyés seront signés pour assurer l'authentification entre les appareils. Les machines cryptent et décryptent les messages avec les nouvelles clés générées. Donc, si on chiffre un message avec R_n, le déchiffrement avec $R_{(n+1)}$ sera refusé. Si un attaquant écoute les données chiffrées avec R_n il ne pourra plus envoyer le même message puisque les nouveaux messages seront cryptés et décryptés par $R_{(n+1)}$.

.2 Évaluation formelle de la sécurité

La vérification formelle est une approche utilisée pour fournir des assurances de sécurité en vérifiant mathématiquement l'exactitude des prototypes à l'aide d'une combinaison de méthodes mathématiques et logiques. Ces approches sont particulièrement utiles pour extraire des informations objectives sur les propriétés de protection et de sécurité des appareils numériques [39].

.3 L'outil « ProVerif »

« ProVerif » est un vérificateur automatique de protocole cryptographique dans le modèle formel (appelé modèle Dolev-Yao). Ce vérificateur de protocole est basé sur une représentation du protocole par des clauses Horn. Ses principales caractéristiques sont :

- Sa capacité à gérer de nombreuses primitives cryptographiques différentes, y compris la cryptographie à clé partagée et publique (chiffrement et signatures), les fonctions de

hachage et les accords de clé Diffie-Hellman, spécifiés à la fois sous forme de règles, de réécriture ou d'équations [40] ;

- Sa capacité à gérer un nombre illimité de sessions du protocole (même en parallèle) et un espace de messages illimité. Ce résultat a été obtenu grâce à des approximations bien choisies. Cela signifie que le vérificateur peut donner de fausses attaques, mais que s'il prétend que le protocole satisfait une propriété, alors la propriété serait en fait satisfaite. L'algorithme de résolution considéré se termine sur une large classe de protocoles (les protocoles dits « tagués »). Lorsque l'outil ne peut pas prouver une propriété, il tente de reconstituer une attaque, c'est-à-dire une trace d'exécution du protocole qui falsifie la propriété recherchée [40].

Aussi, ProVerif peut prouver les propriétés suivantes :

- Secret (l'adversaire ne peut pas obtenir le secret) ;
- Propriétés d'authentification et plus généralement de correspondance ;
- Secret fort (l'adversaire ne voit pas la différence lorsque la valeur du secret change) ;
- Équivalences entre les processus qui ne diffèrent que par des termes.

.4 Scénarios d'attaques

Dans cette section, nous proposons nos différents scénarios d'attaques, suivis par l'interprétation des résultats obtenus.

Dans notre modèle d'attaque (Figure 12), il y a une communication machine à machine avec une clé de *cryptage* ***statique.*** Les paramètres de la clé de chiffrement sont : $\boldsymbol{G}$, $\boldsymbol{Q}$ et $\boldsymbol{P}$ (publiques). Les secrets aléatoires sont : $\boldsymbol{e}$ et $\boldsymbol{d}$ (privés). $\boldsymbol{R}$ est la clé partagée et privée. Une fois que la clé de cryptage a été calculée, les machines entrent dans une communication chiffrée avec la clé partagée. Un attaquant (M?) écoute passivement les données émises par les machines (M1 et M2) sur un canal de communication non sécurisé. Une fois qu'un signal est bien capturé par l'attaquant, ce dernier renvoie le même signal (crypté avec R) à la même machine victime (M2). Lorsque la machine victime (M2) reçoit le signal, elle le décrypte avec sa clé R afin de l'accepter ou de le refuser. Dans notre cas, le signal est crypté avec la même clé R. Donc, la machine victime (M2) accepte le signal et change son état.

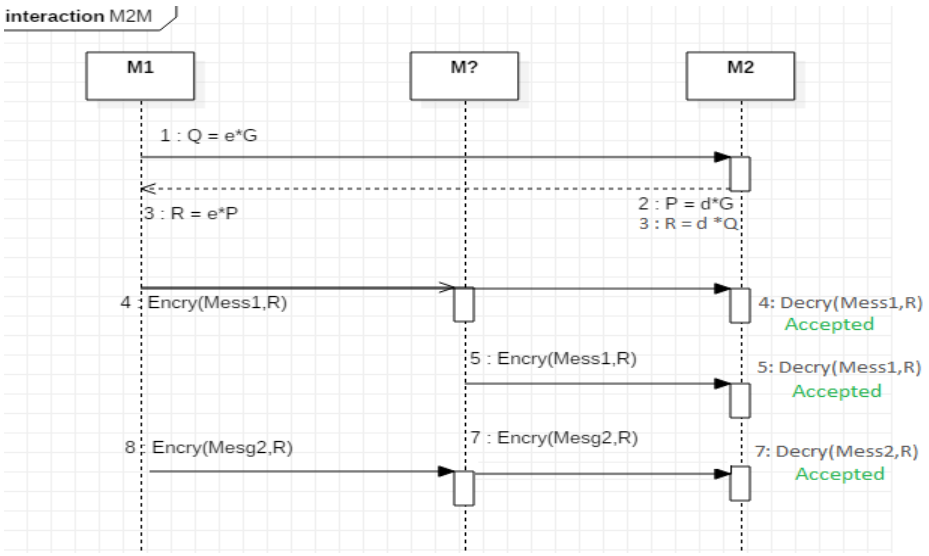


Figure 12 : Interaction M2M avec une clé de cryptage statique

.4.1 Scripts d'attaques

Afin d'évaluer automatiquement la protection du protocole illustré dans le modèle d'attaque à l'aide de l'outil « ProVerif », les explications des codes de calcul (Figure 13) utilisés sont les suivantes :

(1) La valeur « s » est déclarée secrète à l'attaquant en utilisant le mot [private]. « c » est la chaîne publique où « MachineA » et « MachineB» échangent leurs messages. Les fonctions « senc »,« sdec » et « sign » sont respectivement les fonctions de cryptage, de décryptage et de signatures numériques.

(2) La communication M2M est modélisée en définissant quatre évènements qui sont mappés dans les sous-processus « MachineA » et « MachineB » ainsi qu'un ensemble de requêtes.

(3) Dans « process », nous avons utilisé une composition parallèle pour représenter des processus concurrents qui interagissent à l'aide de canaux de communication.

```
(* Symmetric key encryption *)
type key.
fun senc(bitstring, key): bitstring.
reduc forall m: bitstring, k: key; sdec(senc(m,k),k) = m.
(* Asymmetric key encryption *)
type skey.
type pkey.

fun pk(skey): pkey.
fun aenc(bitstring, pkey): bitstring.

reduc forall m: bitstring, sk: skey; adec(aenc(m,pk(sk)),sk) = m.

(* Digital signatures *)

type sskey.
type spkey.

fun spk(sskey): spkey.
fun sign(bitstring, sskey): bitstring.

reduc forall m: bitstring, ssk: sskey; checkMsg(sign(m,ssk),spk(ssk)) = m.

free c:channel.

free s:bitstring [private].
query attacker(s).

event acceptsMachineA(key).
event acceptsMachineB(key,pkey).
event termMachineA(key,pkey).
event termMachineB(key).

query x:key,y:pkey; event(termMachineA(x,y))==>event(acceptsMachineB(x,y)).
query x:key; inj-event(termMachineB(x))==>inj-event(acceptsMachineA(x)).

let MachineA(pkA:pkey,skA:skey,pkB:spkey, G: bitstring ) =
    out(c,pkA);
    in(c,x:bitstring);
    let y = adec(x,skA) in
    let (=pkB,k:key) = checkMsg(y,pkB) in
    event acceptsMachineA(k);
    out(c,senc(s,k));
    event termMachineA(k,pkA).

let MachineB(pkB:spkey,skB:sskey,pkA:pkey,ekB:skey,G: bitstring) =
    in(c,pkX:pkey);
    new k:key;
    event acceptsMachineB(k,pkX);
    out(c,aenc(sign((pkB,k),skB),pkX));
    in(c,x:bitstring);
    let z = sdec(x,k) in
    if pkX = pkA then event termMachineB(k).

process
    new G:bitstring;
    new skA:skey;
    new ekB:skey;
    new skB:sskey;
    let pkA = pk(skA) in out(c,pkA);
    let pkB = spk(skB) in out(c,pkB);
    ( (!MachineA(pkA,skA,pkB,G)) | (!MachineB(pkB,skB,pkA,ekB,G)) )
```

Figure 13 : Scriptes d'attaque

.4.2 Résultats d'exécution

« ProVerif » est une méthode dans laquelle un ensemble de commandes est utilisé pour évaluer automatiquement les protocoles cryptographiques sur la base de la description par les règles « Prolog ». Sur l'invité de commande, pour exécuter le scripte illustré dans la figure 13, on utilise la commande : ./proverif M2M_Model_Attack.pv

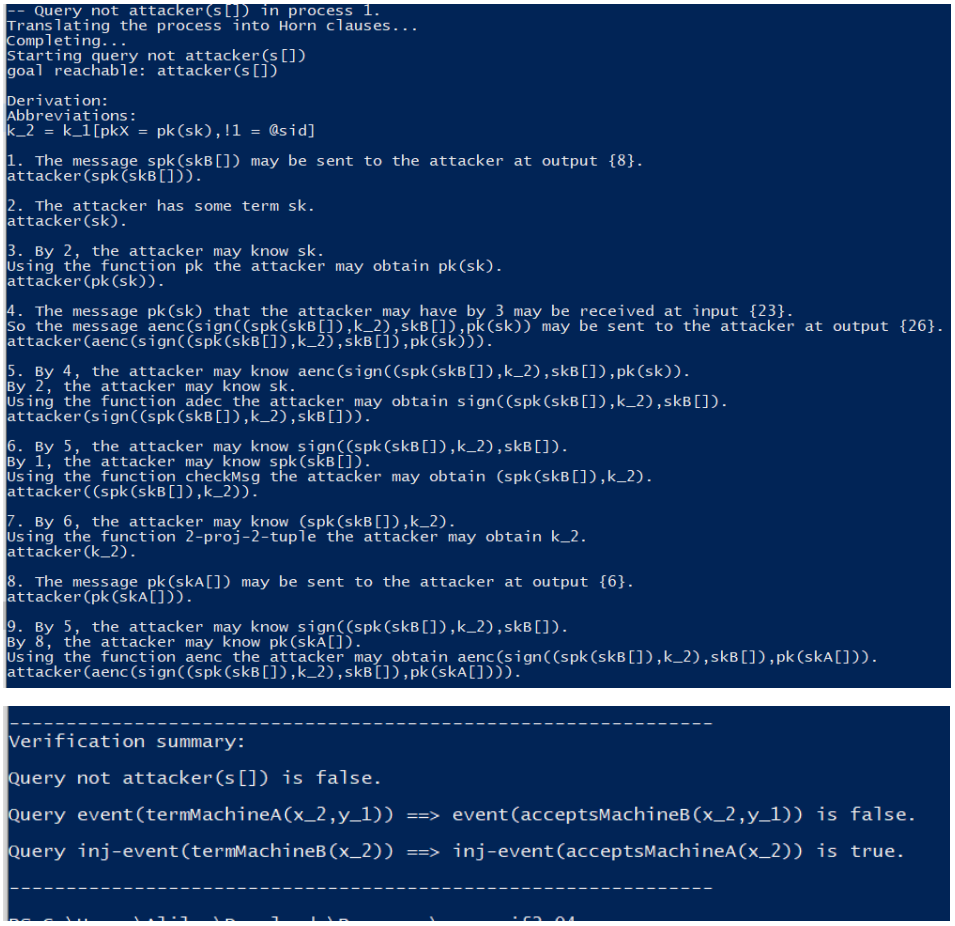


Figure 14 : Résultats de la dérivation et traces d'attaque

Dans le résumé de vérification illustré à la figure 14, il est possible de constater qu'une attaque peut être dérivée, ce qui est logique puisqu'elle est envoyée dans le canal public. Il est nécessaire de considérer la différence entre la dérivation d'attaque et la trace d'attaque afin d'interpréter correctement les résultats. La dérivation d'attaque est une description du comportement que l'attaquant doit adopter pour violer la propriété de protection dans la représentation interne de « ProVerif ». Quant à elle, la trace d'attaque dans notre modèle est vue juste avant la « verification summary » par les résultats de 1 jusqu'à 9 (Figure 14).

- `RÉSULTAT [Query not attacker is false]` : « ProVerif » a découvert une intrusion sur la propriété de protection requise.
- `RÉSULTAT[Queryevent(termMachineA(x_2,y_1))==>event(acceptsMachine B(x_2,y_1))is false]` *:* « ProVerif » a découvert qu'un message envoyé par la « MachineA » et accepté par la « MachineB » n'est toujours pas vérifié.

Les résultats d'exécution montrent que la communication est vulnérable à certains types d'attaques d'écoute. La machine A peut envoyer à un attaquant un message ou une clé publique. Avec ces deux éléments, l'attaquant est capable de recevoir les données ou de les envoyer à ces machines afin de changer leurs états.

.5 Contre-Mesures

Dans le modèle de contre-mesures (Figure 15), on a utilisé une communication machine à machine avec une génération **dynamique** de la clé de *chiffrement*. Les paramètres de la clé de chiffrement sont : $\boldsymbol{G}$, $\boldsymbol{Q}$ et $\boldsymbol{P}$ (publiques). Les secrets aléatoires sont : $\boldsymbol{e}$ et $\boldsymbol{d}$ (privés). Les clés générées R sont définies par $\boldsymbol{R_1}$, ... , $\boldsymbol{R_n}$: (privées aussi).

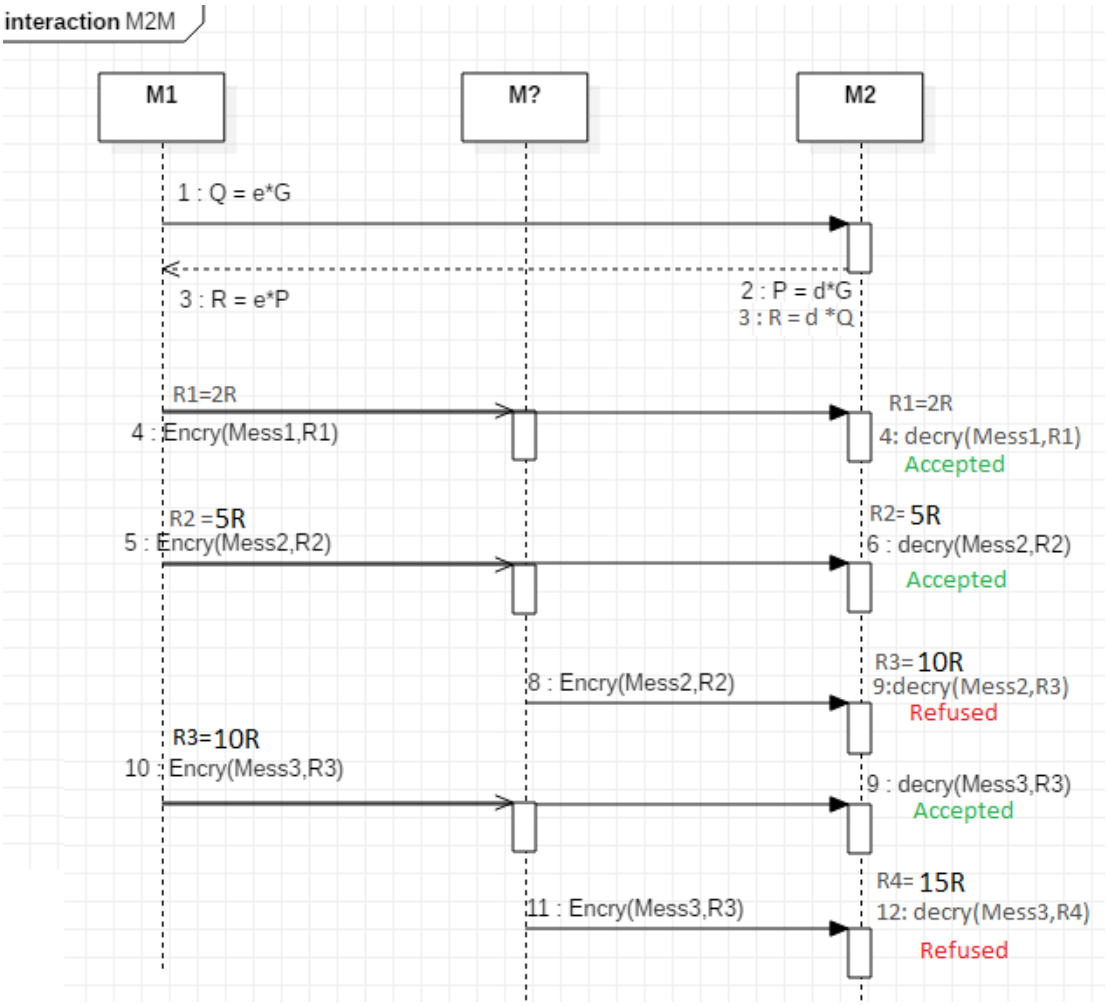


Figure 15 : Interaction M2M avec une clé de cryptage dynamique

Les machines génèrent une nouvelle clé de cryptage et de décryptage après chaque transaction valide. L'ancienne clé de cryptage sera expirée, car la clé de cryptage et de décryptage R1 est remplacée par R2. Donc, si l'attaquant (M?) écoute passivement les signaux émis par les machines (M1 et M2), il renvoie le même signal crypté avec R1 à la machine victime (M2). Le signal sera alors refusé, car la clé de décryptage de la machine victime est changée à R2. De cette façon, la réutilisation d'un signal ne sera plus acceptée, et la non-répudiation, l'anonymat et l'authentification seront assurés.

.5.1 Scripts de contre-mesures

Dans le script de contre-mesures (Figure 16), on a utilisé les mêmes paramètres que dans le script d'avant et on a ajouté les fonctions de génération des clés comme suit :

(1) Les fonctions « ecc », « add » et « newKey » représentent respectivement les paramètres de la clé de chiffrement ECC, l'addition de deux points ECC ainsi que la génération d'une nouvelle clé de chiffrement.

(2) La communication M2M est modélisée en définissant quatre évènements qui sont « mappés » dans les sous-processus « MachineA » et « MachineB» ainsi que des ensembles de requêtes pour la génération des clés.

(3) Dans « process », nous avons utilisé la même composition parallèle pour représenter les processus concurrents qui interagissent à l'aide de canaux de communication.

```
(* Symmetric key encryption *)
type key.
fun senc(bitstring, key): bitstring.
reduc forall m: bitstring, k: key; sdec(senc(m,k),k) = m.

(* Asymmetric key encryption *)
type skey.
type pkey.
fun pk(skey): pkey.
fun aenc(bitstring, pkey): bitstring.
reduc forall m: bitstring, sk: skey; adec(aenc(m,pk(sk)),sk) = m.

(* Digital signatures *)
type sskey.
type spkey.
fun spk(sskey): spkey.
fun sign(bitstring, sskey): bitstring.

reduc forall m: bitstring, ssk: sskey; checkMsg(sign(m,ssk),spk(ssk)) = m.
fun mecc(skey, bitstring):skey.
fun add (bitstring, bitstring):bitstring.
fun newKey (bitstring): bitstring.
equation forall x:bitstring, y:bitstring; newKey(x)=add(x,x).

free c:channel.
free s:bitstring [private].
query attacker(s).

event acceptsMachineA(key).
event acceptsMachineB(key,pkey).
event termMachineA(key,pkey).
event termMachineB(key).

query x:key,y:pkey; event(termMachineA(x,y))==>event(acceptsMachineB(x,y)).
query x:key; inj-event(termMachineB(x))==>inj-event(acceptsMachineA(x)).

let MachineA(pkA:pkey,skA:skey,pkB:spkey, G: bitstring ) =
    out(c,pkA);
    in(c,x:bitstring);
    let RR=mecc(skA, G) in
    new sr:skey;
    let y = adec(x,sr) in
    let (=pkB,k:key) = checkMsg(y,pkB) in
    event acceptsMachineA(k);
    out(c,senc(s,k));
    event termMachineA(k,pkA).

let MachineB(pkB:spkey,skB:sskey,pkA:pkey,ekB:skey,G: bitstring) =
    in(c,pkX:pkey);
    new k:key;
    let RR=mecc(ekB, G) in
    new sr:skey;
    event acceptsMachineB(k,pkX);
    out(c,aenc(sign((pkB,k),skB),pkX));
    in(c,x:bitstring);
    let z = sdec(x,k) in
    if pkX = pkA then event termMachineB(k).

process
    new G:bitstring;
    new skA:skey;
    new ekB:skey;
    new skB:sskey;
    let pkA = pk(skA) in out(c,pkA);
    let pkB = spk(skB) in out(c,pkB);
    ( (!MachineA(pkA,skA,pkB,G)) | (!MachineB(pkB,skB,pkA,ekB,G)) )
```

Figure 16 : Scriptes de contre-mesures

.5.2 Résultats d'exécution

Dans le résumé de vérification, illustré à la figure 17, on peut constater que la requête a été confirmée et qu'il n'y a pas de menace. Dans ce scénario, « ProVerif » ne révèle aucune dérivation d'attaque et aucune trace d'attaque.

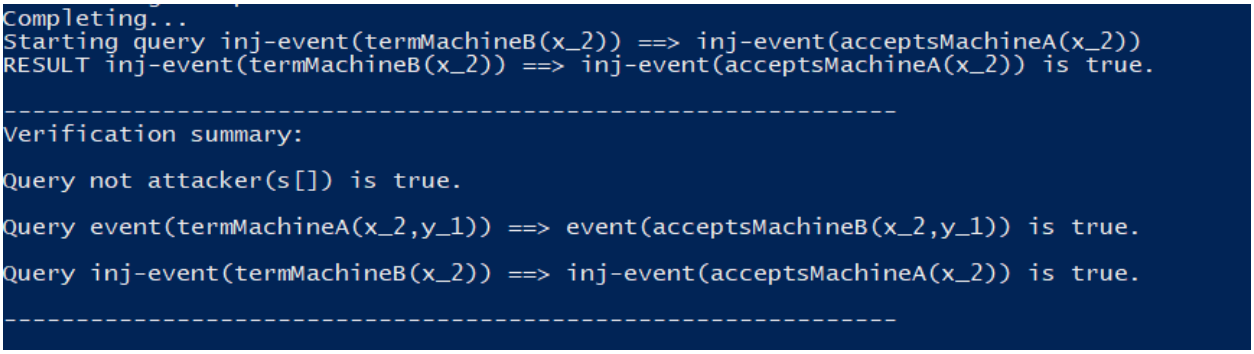


Figure 17 : Résultats de vérification de la trace d'attaque

Les résultats d'exécution montrent que la communication n'est plus vulnérable aux différents types d'attaques d'écoute. De plus, malgré qu'un attaquant soit capable d'écouter un message ou une clé publique émis sur un canal de communication, l'attaquant n'est plus capable d'envoyer des données valides à ces machines.

.6 Évaluation de la sécurité du protocole

Notre protocole répond aux exigences de sécurité informatique telles que :

- **L'anonymat / La non-répudiation**

Dans notre modèle, les communications M2M sont cryptées et décryptées avec de nouvelles clés générées dynamiquement, ce qui bloquent différents types d'attaques telles que les attaques d'écoute et d'injection des messages.

- **L'Authentification / La confidentialité**

Notre modèle est renforcé par les signatures numériques de la cryptographie ECC, ce qui offre un niveau amélioré d'authentification et de confidentialité aux communications numériques. De plus, au niveau de l'architecture globale du système, le modèle de publication et d'abonnement du protocole MQTT assure la confidentialité des informations pour que les données ne soient accessibles qu'aux machines autorisées.

- **L’intégrité**

Dans notre modèle, les données sont distribuées entre les « Brokers » MQTT et le système ne comporte pas de point de défaillance unique. La falsification des données nécessite l’attaque de tous les nœuds du réseau.

.7 Conclusion

Sur la base de l’évaluation formelle et des scénarios d’attaques, on peut conclure que l’utilisation d’une clé de chiffrement statique entraine des attaques d’écoute et d’injection par des entités malveillantes. Cependant, la génération dynamique des clés de chiffrement assure la sécurité des informations transmises entre les machines, l’authentification des appareils, l’anonymat et la non-répudiation.

Dans le chapitre suivant, nous présentons les différents scénarios de simulation ainsi que l’analyse et l’interprétation des résultats obtenus afin de mesurer l’efficacité de notre solution.

CHAPITRE 5 SIMULATION ET ÉVALUATION DES PERFORMANCES

L'objectif principal des simulations est d'analyser quels architectures et protocoles d'application sont les plus efficaces en termes de consommation d'énergie et de nombre de paquets échangés. Dans cette partie, nous présentons l'outil de simulation « Contiki Cooja », les paramètres de simulation, l'analyse et l'interprétation des résultats de chaque scénario simulé, et finalement, l'analyse comparative des trois architectures (centralisée, décentralisée, notre solution) simulées afin de voir l'amélioration réalisée.

.1 Outils de simulation « Contiki Cooja »

Afin d'évaluer les performances des protocoles d'application sur les différentes architectures en termes de consommation d'énergie et de trafic généré, on a utilisé « Contiki », un système d'exploitation « open source » spécialement conçu pour l'IoT. « Contiki » possède les fonctionnalités du noyau de lecteur d'évènements et du « multithreading » préemptif. « Contiki » prend entièrement en charge les protocoles traditionnels de la pile de protocole Internet (IP) tels que l'« User Datagram Protocol (UDP) », le « Transmission Control Protocol (TCP) » et l'« Hypertext Transfer Protocol (HTTP) » ainsi que les normes de l'IoT telles que les réseaux personnels sans fil à faible consommation (6LowPAN), le « Routing protocol for Low-Power (RPL) », « The Constrained Application Protocol (CoAP) » et le « Message Queuing Telemetry Transport (MQTT) ». Toutes les applications de « Contiki » sont écrites en langage de programmation C. « Contiki » dispose d'un outil très puissant, appelé « Cooja », qui est un outil d'analyse des données et un simulateur où les réseaux peuvent être testés avant d'être déployés dans le matériel. « Cooja » offre l'avantage de voir à tout moment ce qui se passe dans le réseau simulé et permet aux développeurs de déboguer facilement les logiciels pour de tels réseaux, ce qui est assez difficile [41]. « Contiki » est conçu pour fonctionner avec des appareils à faible puissance, par exemple, des appareils dont la batterie peut durer des années. « Contiki » fournit des mécanismes pour estimer la consommation d'énergie du système et pour comprendre où l'énergie a été consommée.

.2 Paramètres de simulations

Dans notre simulation, on a implémenté les paramètres décrits dans le Tableau 1 et calculé la consommation d'énergie par chaque nœud dans les différentes architectures réseau. Le temps de simulation est de 90 minutes, un temps suffisamment long pour analyser ce qui se passe avec les différentes architectures.

Paramètre	**Valeur**
Radio medium	Unit Disk Graph Medium
MAC layer	CSMA/CA
Wireless protocol	802.15.4 Radio
Transport layer	TCP
Application Layer	MQTT-SN
Total node	18
Node transmission range	50 meters
Tx/Rx ratio	100%
Mote type	T-mote sky
Simulation Time	90 minutes

Tableau 1 : Paramètres de simulations

Chacun des nœuds a implémenté « l'Unit Disk Graph Medium (UDGM) » et le « Carrier Sense Multiple Access (CSMA/CA) ». L'UDGM n'est utilisé que lorsqu'un réseau homogène est créé. L'UDGM contient toutes les informations liées à la topologie du réseau et les emplacements des nœuds.

Au niveau de la couche « application », on a utilisé le protocole « MQTT for sensor networks (MQTT_SN) ».

Le « Node Transmission range » présente la plage de transmission par nœud ainsi que le « Tx/Rx » présente respectivement la plage d'interférence et le rapport de transmission par nœud [42].

Les « T- motes sky », des nœuds de « Contiki », utilisent des microcontrôleurs MSP430 [41] qui sont conçus pour fonctionner avec de petites quantités de mémoire.

Le simulateur « Cooja » surveille l'utilisation de l'énergie dans le nœud du capteur via le module complémentaire « Powertrace ». La mesure de la consommation d'énergie dans le simulateur « Cooja » est basée sur quatre paramètres :

- La puissance du processeur (CPU) : fait référence à l'énergie de calcul requise par le nœud ;
- La puissance du mode basse consommation (LPM) : fait référence à l'énergie utilisée lorsque le nœud du capteur est à l'état inactif ;

- La puissance d'écoute (Radio Listen) : fait référence à l'énergie requise lorsque le nœud du capteur est prêt à recevoir le paquet de données de ses nœuds voisins ;
- La puissance de transmission (Radio Transmit) : fait référence à l'énergie requise par le nœud du capteur pour transmettre le paquet de données à ses voisins.

.3 Scénarios de simulation

.3.1 Simulation de l'architecture centralisée

La figure 18 présente une architecture centralisée basée sur un seul contrôleur central (nœud1). Tous les autres nœuds (nœud 2...13) échangent leurs données selon le modèle de publication et d'abonnement du protocole MQTT ainsi que selon les paramètres de cryptage ECC. Dans cette architecture, on a implémenté une configuration de réseau dans laquelle les nœuds doivent communiquer avec un seul contrôleur central pour pouvoir communiquer entre eux. Le simulateur « Cooja » nous permet de voir tout le trafic réseau (flèche bleue) ainsi que les messages échangés entre nœuds sur la console.

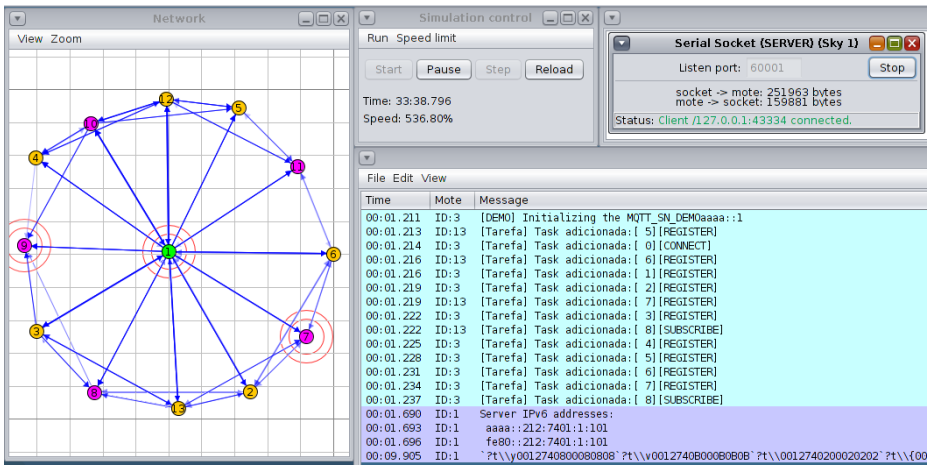
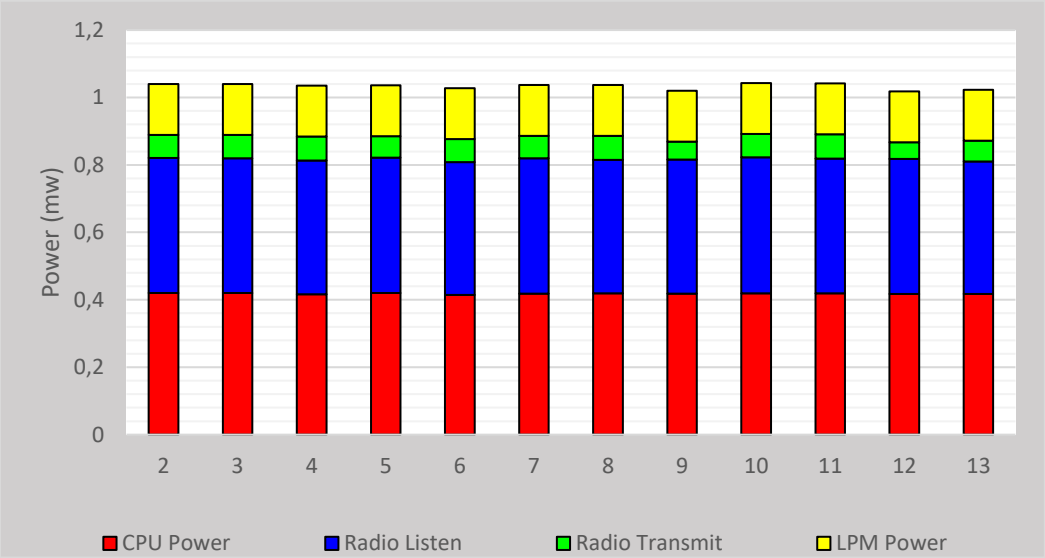


Figure 18 : Simulation de l'architecture centralisée

L'histogramme 1 et le tableau 2 montrent les résultats de consommation d'énergie par les nœuds en termes de CPU, de Radio Listen, de Radio Transmit et de LPM. La dernière ligne du tableau 2 présente la consommation d'énergie moyenne pour tous les nœuds. On peut observer que, pour les 235 paquets reçus, lorsque les nœuds ont été actifs pendant 43 minutes et 32 secondes, la consommation d'énergie par CPU (0,418 mw) et par Radio Listen (0,399 mw) est la plus couteuse. La LPM est constante à environ 0.0151

mégawatt (mw) et la moyenne générale de consommation d'énergie est de 1,033 mw par nœud.



Histogramme 1 : Consommation moyenne d'énergie (architecture centralisée)

Noeud	Paquets reçus	Temps d'activité	CPU Power	LPM Power	Radio Listen	Radio Transmit	Énergie totale
2	235	43 min. 24 sec	0,42	0,151	0,401	0,068	1,04
3	236	43 min. 32 sec	0,42	0,151	0,4	0,069	1,04
4	235	43 min. 32 sec	0,416	0,151	0,397	0,071	1,035
5	235	43 min. 00 sec	0,42	0,151	0,402	0,063	1,036
6	236	42 min. 58 sec	0,414	0,151	0,394	0,069	1,028
7	235	44 min. 05 sec	0,418	0,151	0,402	0,066	1,037
8	236	42 min. 51 sec	0,419	0,151	0,396	0,071	1,037
9	235	42 min. 25 sec	0,418	0,151	0,398	0,053	1,02
10	235	43 min. 02 sec	0,419	0,151	0,404	0,069	1,043
11	236	42 min. 12 sec	0,419	0,151	0,4	0,072	1,042
12	236	43 min. 00 sec	0,417	0,151	0,401	0,049	1,018
13	235	43 min. 21 sec	0,417	0,151	0,393	0,062	1,023
AVG	235	43 min. 32 sec	0,418	0,151	0,399	0,065	1,033

Tableau 2 : Consommation moyenne d'énergie par nœud (architecture centralisée)

.3.2 Simulation de l'architecture décentralisée

La figure 19 présente une architecture totalement décentralisée, le nœud 1 est la passerelle de l'IoT, alors que tous les autres nœuds (nœud 2...15) échangent leurs données d'une manière « Peer to Peer (P2P) ». Dans cette architecture, un groupe d'appareils échangent collectivement des données et chaque nœud agit comme un pair individuel. Le simulateur « Cooja » nous permet de voir en tout temps la consommation d'énergie instantanée (histogramme 2) par nœud, leur trafic réseau et les messages échangés.

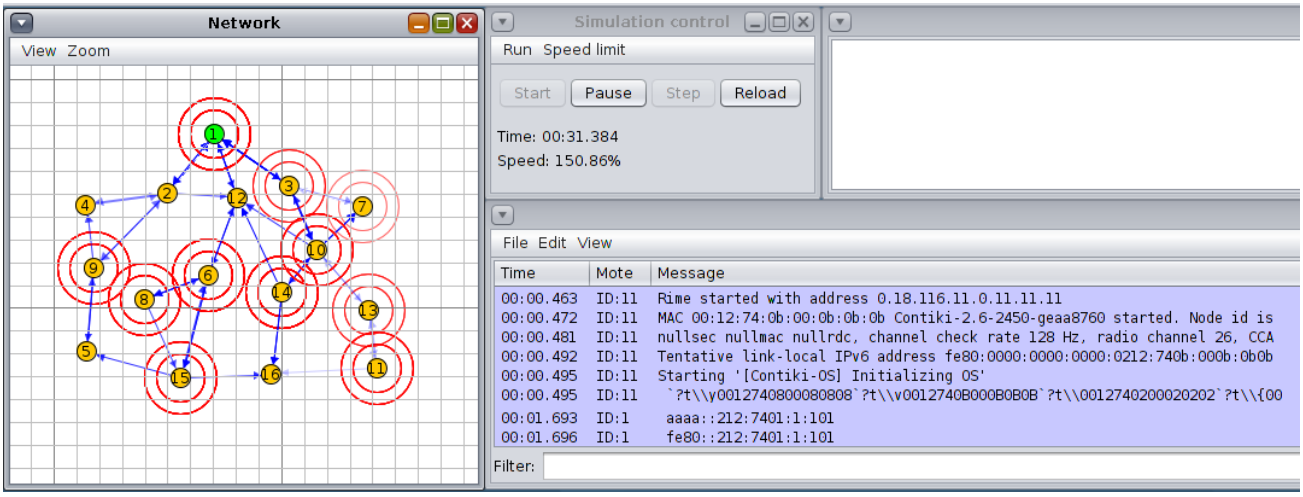
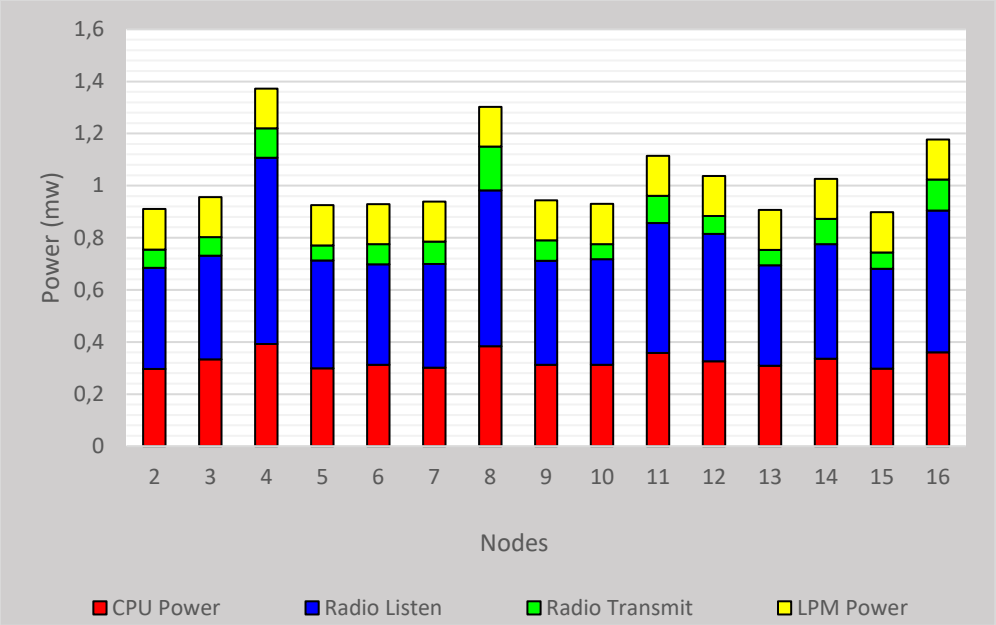


Figure 19 : Architecture totalement décentralisée

L'histogramme 2 présente les résultats de consommation d'énergie en termes de CPU, de Radio Listen, de Radio Transmit et de LPM. La consommation d'énergie par CPU et par Radio Listen power est la plus couteuse. Le LPM est presque constant (0,154 mw). Les nœuds 4, 8 et 11 sont les plus énergivores en raison du nombre de paquets reçus. La moyenne générale de consommation d'énergie est de 1,024 mw par nœud.



Histogramme 2 : Consommation moyenne d'énergie (architecture décentralisée)

Le tableau 3 donne plus de détails sur le nombre de messages reçus et sur la consommation d'énergie par chaque nœud. À travers ce tableau, on peut analyser la moyenne de consommation d'énergie par tous les nœuds pour une moyenne de 235 paquets reçus, par exemple le nœud 4 a consommé une énergie moyenne de 1,372 mw pour 285 messages reçus lorsque le nœud a été actif pendant 43 minutes et 32 secondes.

Noeud	Paquets reçus	Temps d'activité	CPU Power	LPM Power	Radio Listen	Radio Transmit	Énergie totale
2	235	43 min. 24 sec	0,297	0,155	0,387	0,071	0,91
3	236	43 min. 32 sec	0,333	0,153	0,398	0,072	0,956
4	285	43 min. 32 sec	0,392	0,152	0,715	0,113	1,372
5	235	43 min. 00 sec	0,299	0,154	0,414	0,058	0,925
6	255	42 min. 58 sec	0,312	0,154	0,386	0,077	0,929
7	256	44 min. 05 sec	0,301	0,154	0,398	0,086	0,939
8	275	42 min. 51 sec	0,384	0,152	0,598	0,168	1,302
9	255	42 min. 25 sec	0,313	0,154	0,398	0,079	0,944
10	236	43 min. 02 sec	0,312	0,154	0,406	0,058	0,93
11	265	42 min. 12 sec	0,358	0,153	0,499	0,104	1,114
12	236	43 min. 00 sec	0,326	0,154	0,489	0,068	1,037
13	235	43 min. 21 sec	0,309	0,154	0,385	0,059	0,907
14	235	42 min. 25 sec	0,336	0,153	0,439	0,098	1,026
15	236	43 min. 02 sec	0,298	0,154	0,383	0,063	0,898
16	234	42 min. 12 sec	0,36	0,153	0,545	0,119	1,177
Avg	235	42 min. 25 sec	0,329	0,154	0,456	0,086	1,024

Tableau 3 : Consommation moyenne d'énergie par nœud (architecture décentralisée)

.3.3 Simulation de notre architecture avec les mécanismes de sécurité

La figure 20 présente notre architecture réseau. Les nœuds 1, 2 et 3 sont des « Brokers » MQTT, alors que les autres nœuds (nœud 4...18) sont des clients MQTT qui font les échangent de leurs données de manière P2P. De plus, chaque nœud est associé à un « Broker » MQTT spécifique afin d'implémenter le modèle de publication et d'abonnement du protocole MQTT ainsi que notre mécanisme de sécurité. Le simulateur « Cooja » nous permet de voir tout le trafic réseau ainsi que les messages échangés.

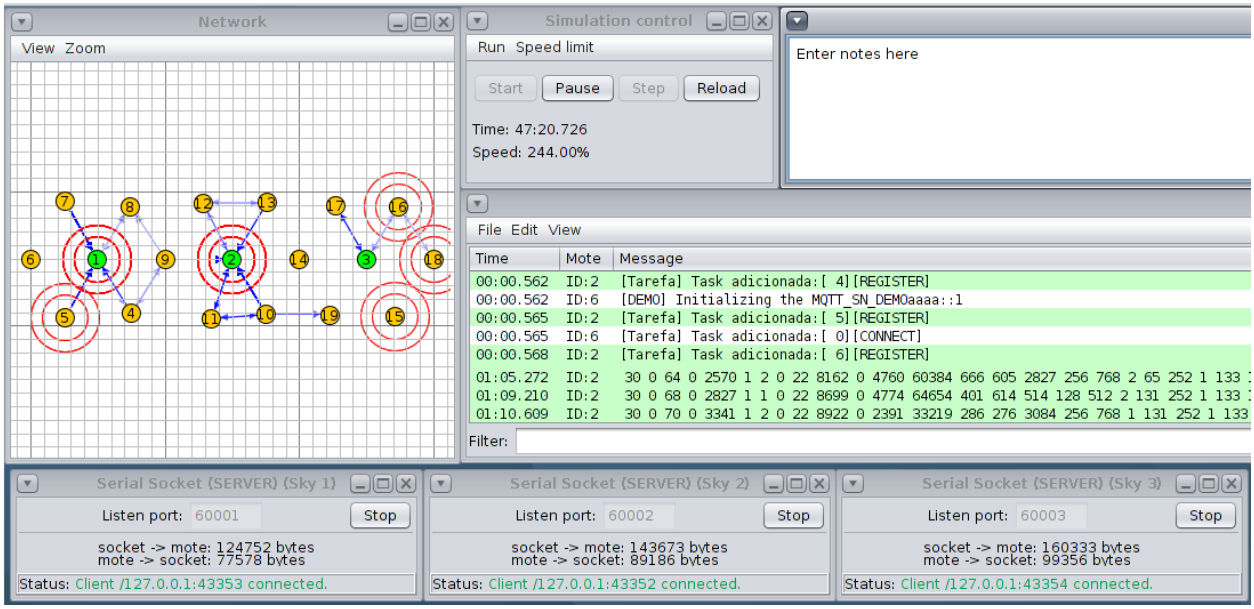
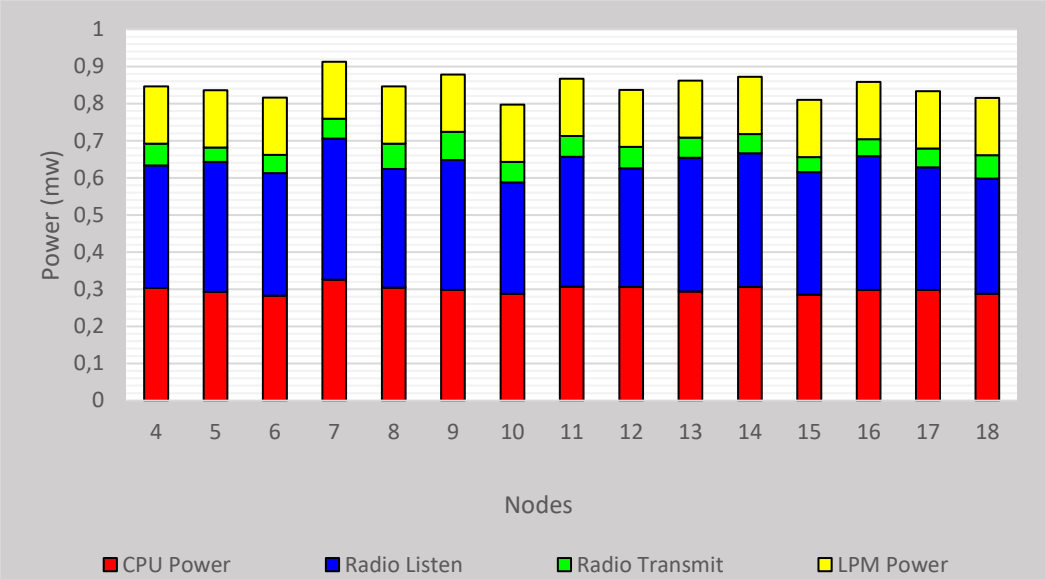


Figure 20 : Notre architecture avec les mécanismes de sécurité

L'histogramme 3 montre les résultats de consommation d'énergie en termes de CPU, de Radio Listen, de Radio Transmit et de LPM dans notre modèle. Le tableau 4 présente la moyenne de consommation d'énergie par tous les nœuds pour un nombre moyen de 235 paquets reçus par nœud. La moyenne de consommation d'énergie par le CPU est de 0,298 mw et de 0,339 mw par Radio Listen. Le LPM est toujours constant (0,154 mw) et la moyenne générale de consommation d'énergie par chaque nœud est de 0,815 mw.



Histogramme 3 : Consommation moyenne d'énergie (notre architecture avec les mécanismes de sécurité)

Noeud	Paquets reçus	Temps d'activité	CPU Power	LPM Power	Radio Listen	Radio Transmit	Énergie totale
4	235	44 min. 05 sec	0,303	0,154	0,33	0,059	0,846
5	236	44 min. 05 sec	0,293	0,154	0,35	0,039	0,836
6	285	43 min. 24 sec	0,283	0,154	0,33	0,049	0,816
7	235	43 min. 32 sec	0,326	0,154	0,38	0,053	0,913
8	255	43 min. 32 sec	0,304	0,154	0,32	0,068	0,846
9	256	43 min. 00 sec	0,298	0,154	0,35	0,076	0,878
10	275	42 min. 58 sec	0,288	0,154	0,3	0,055	0,797
11	255	44 min. 05 sec	0,307	0,154	0,35	0,056	0,867
12	236	42 min. 51 sec	0,306	0,154	0,32	0,057	0,837
13	265	42 min. 25 sec	0,294	0,154	0,36	0,054	0,862
14	236	43 min. 02 sec	0,306	0,154	0,36	0,052	0,872
15	235	42 min. 12 sec	0,285	0,154	0,33	0,041	0,81
16	235	43 min. 00 sec	0,298	0,154	0,36	0,046	0,858
17	236	43 min. 21 sec	0,298	0,154	0,33	0,051	0,833
18	234	44 min. 05 sec	0,288	0,154	0,31	0,063	0,815
AVG	232	43 min. 32 sec	0,298	0,154	0,339	0,055	0,846

Tableau 4 : Consommation moyenne d'énergie par nœud (notre architecture avec les mécanismes de sécurtié)

.3.4 Simulation de notre architecture sans mécanismes de sécurité

La figure 21 présente notre architecture réseau sans mécanismes de sécurité. Les nœuds 1, 2 et 3 sont des « Brokers » MQTT, alors que les autres nœuds (nœud 4...18) sont des clients MQTT qui échangent leurs données de manière P2P. Le simulateur « Cooja » nous permet de voir tout le trafic réseau ainsi que les messages échangés.

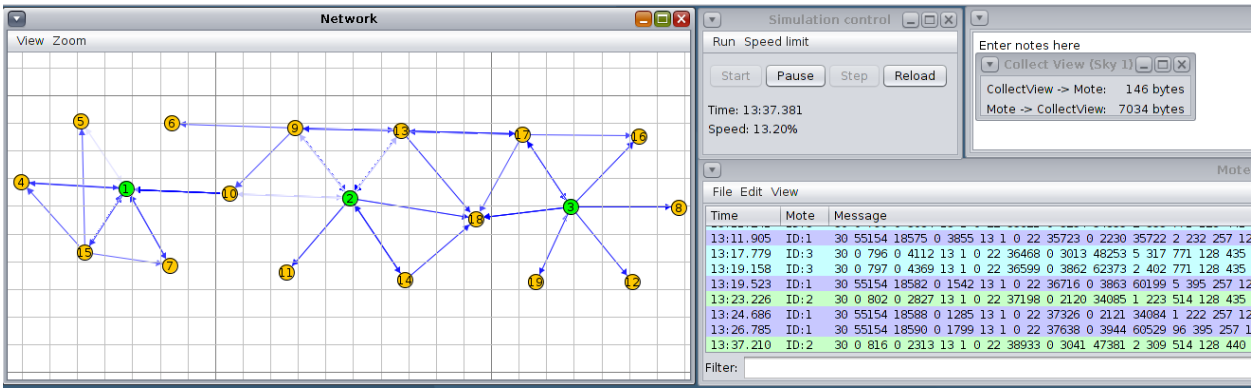
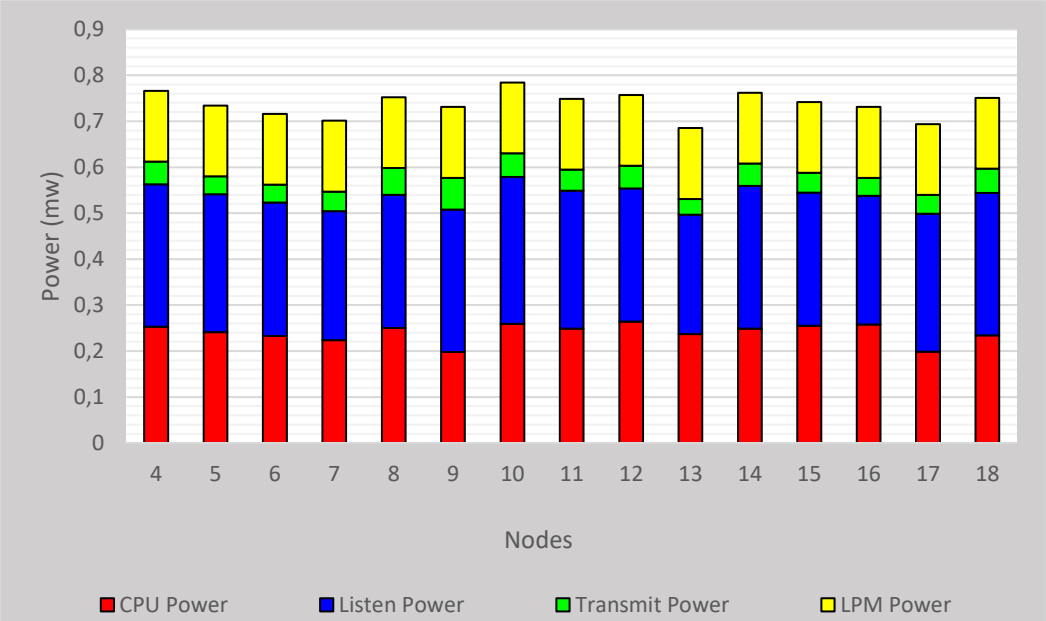


Figure 21: Notre architecture sans mécanismes de sécurité

L'histogramme 4 montre les résultats de consommation d'énergie en termes de CPU, de Radio Listen, de Radio Transmit et de LPM dans notre modèle sans mécanismes de sécurité. Le tableau 5 présente la moyenne de consommation d'énergie par tous les nœuds pour un nombre moyen de 231 paquets reçus par nœud. La moyenne de consommation d'énergie par le CPU est de 0,240 mw et de 0,296 mw par Radio Listen. Le LPM est toujours constant (0,154 mw) et la moyenne générale de consommation d'énergie par chaque nœud est de 0,737 mw.



Histogramme 4 : Consommation moyenne d'énergie (notre architecture sans mécanismes de sécurtié)

Noeud	Paquets reçus	Temps d'activité	CPU Power	LPM Power	Radio Listen	Radio Transmit	Énergie totale
4	222	41 min. 05 sec	0,253	0,154	0,31	0,049	0,766
5	225	42 min. 07 sec	0,241	0,154	0,3	0,039	0,734
6	230	42 min. 25 sec	0,233	0,154	0,29	0,039	0,716
7	221	43 min. 22 sec	0,224	0,154	0,28	0,043	0,701
8	230	44 min. 23 sec	0,25	0,154	0,29	0,058	0,752
9	233	44 min. 02 sec	0,198	0,154	0,31	0,069	0,731
10	229	40 min. 34 sec	0,259	0,154	0,32	0,051	0,784
11	235	41 min. 04 sec	0,249	0,154	0,3	0,046	0,749
12	233	43 min. 23 sec	0,264	0,154	0,29	0,049	0,757
13	240	42 min. 26 sec	0,237	0,154	0,26	0,034	0,685
14	232	44 min. 12 sec	0,249	0,154	0,31	0,049	0,762
15	234	41 min. 22 sec	0,255	0,154	0,29	0,043	0,742
16	235	40 min. 20 sec	0,258	0,154	0,28	0,039	0,731
17	236	41 min. 24 sec	0,199	0,154	0,3	0,041	0,694
18	230	44 min. 25 sec	0,234	0,154	0,31	0,053	0,751
AVG	231	43 min. 32 sec	0,240	0,154	0,296	0,047	0,737

Tableau 5 : Consommation moyenne d'énergie par nœud (notre architecture sans mécanismes de sécurité)

.4 Analyse comparative

Les courbes présentées à la figure 22 montrent la consommation d'énergie moyenne en termes de CPU, de LPM, de Radio Listen et de Radio Transmit dans les divers modèles de simulation.

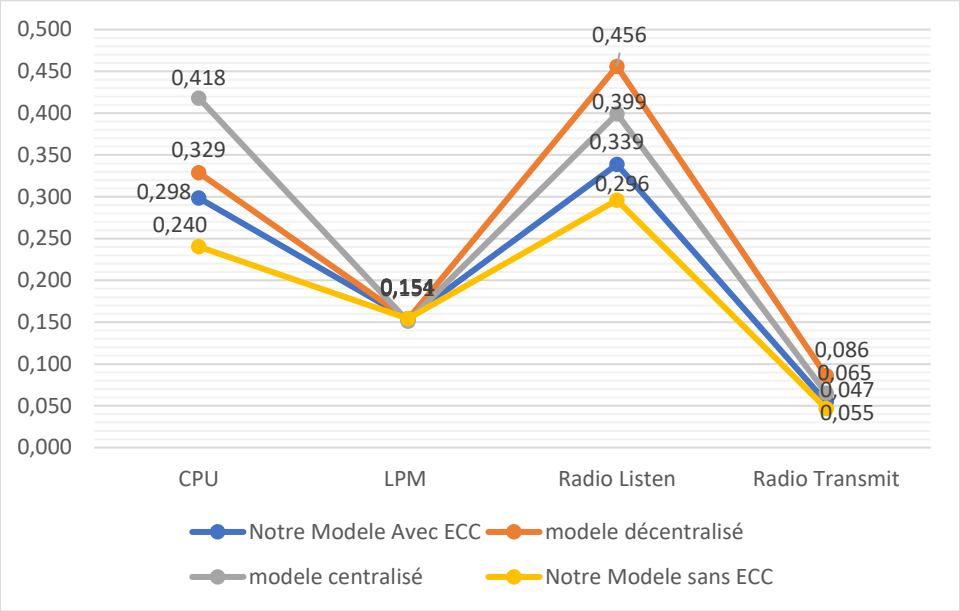


Figure 22 : Consommation d'énergie moyenne des architectures réseau

Après avoir calculé la moyenne de consommation d'énergie de chaque nœud dans les différentes architectures réseaux, le tableau 6 résume les résultats obtenus dans les différents modèles simulés ainsi que l'amélioration réalisé par notre modèle.

Consommation d'énergie	CPU	LPM	Radio Listen	Radio Transmit	Totale
Modèle centraliser (mw)	0,418	0,151	0,399	0,065	1,033
Modèle décentraliser (mw)	0,329	0,154	0,456	0,086	1,025
Notre Modèle avec les mécanismes de sécurité (mw)	0,298	0,154	0,339	0,055	0,846
Notre modèle sans mécanismes de sécurité (mw)	0,240	0.154	0,296	0,047	0,737
Améliorerions par rapport au modèle centraliser (%)	28,708 %	-1,987 %	15,038 %	15,385 %	18,103 %
Améliorerions par rapport au modèle décentraliser (%)	9,422 %	0,000 %	25,658 %	36,047 %	17,463 %

Tableau 6 : Résultats et améliorations réalisées

À travers ces résultats, on peut constater que la consommation d'énergie par le **CPU** dans l'architecture centralisée est plus coûteuse que l'architecture décentralisée. Dans notre architecture, nous avons réduit le coût de calcul pour le **CPU** de 28% par rapport à l'architecture centralisée et de 9% par rapport à l'architecture décentralisée. Le coût d'implémentation des mécanismes de sécurité par le **CPU** dans notre architecture est de 0,058 mw.

L'énergie utilisée lorsque le nœud du capteur était à l'état inactif (**LPM**) est presque identique pour les trois architectures (≈0,154).

L'énergie requise lorsque le nœud du capteur était prêt à recevoir le paquet de données de ses nœuds voisins (**Radio Listen**) est plus coûteuse dans l'architecture décentralisée que dans l'architecture centralisée. Dans notre architecture, le Radio Listen est réduit de 15% par rapport à l'architecture centralisée et de 25% par rapport à l'architecture décentralisée. Le coût d'implémentation des mécanismes de sécurité par le **Radio Listen** dans notre architecture est de 0,043 mw.

L'énergie requise par le nœud du capteur pour transmettre le paquet de données à ses voisins (**Radio Transmit**) dans l'architecture décentralisée est de 0 ,086 mw plus élevée que dans l'architecture centralisée (0,065 mw). Dans notre architecture, nous avons réduit le coût de calcul pour le **Radio Transmit** à 0,055 mw, soit une réduction de 15% par rapport à l'architecture centralisée et de 36% par rapport à l'architecture décentralisée. Le coût d'implémentation des mécanismes de sécurité par le **Radio Transmit** dans notre architecture est de 0,008 mw.

Le coût total d'implémentation des mécanismes de sécurité dans notre architecture est de 0,109 mw.

.5 Conclusion

Sur la base des scénarios de simulation, on peut conclure que l'architecture réseau, plus précisément le protocole de communication et les mécanismes de sécurité, a un impact significatif sur la consommation d'énergie du nœud du capteur en termes de **CPU**, **Radio Listen** et **Radio Tansmit**. Cependant, il n'existe pas de différences significatives dans la consommation d'énergie lorsque le nœud du capteur était à l'état inactif pour les différentes architectures réseau. Notre modèle partage l'accès et l'utilisation de ressources sur plusieurs contrôleurs, ce qui réduit le coût de calcul sur un seul mineur central. De plus, l'utilisation des mécanismes de sécurité basés sur le ECC repose sur l'utilisation de clés plus petites, ce qui réduit à la fois l'utilisation de la mémoire

et la surcharge de communication. Afin d'obtenir une sécurité de 128 bits, une taille de champ de 256 bits pour le ECC est suffisante. Cependant, le RSA nécessite au moins 3072 bits pour le même niveau de sécurité. La génération et la mise à jour automatique des clés de chiffrements réduisent également le nombre de transactions M2M ainsi que le coût de calcul par ces mineurs.

CHAPITRE 6 CONCLUSION GÉNÉRALE

La sécurité de l'IoT a suscité beaucoup d'attention ces dernières années. Cependant, les problèmes de sécurité liés aux objets connectés posent de nouveaux défis en lien avec les différentes architectures de l'IoT à la maison. Dans notre état de l'art, nous nous sommes concentrés sur les différents problèmes de sécurité et nous avons présenté plusieurs études qui ont proposé différentes solutions pour augmenter la sécurité de l'Internet des objets

Nous avons examiné une variété de défis de sécurité pertinents pour la maison intelligente et nous avons mis en place une maison intelligente fiable en appliquant une architecture flexible qui combine l'architecture décentralisée P2P ainsi que l'architecture centralisée pour gérer les transactions et les flux de données dans le domaine de l'IoT à la maison. Notre architecture repose sur l'utilisation de petites clés, ce qui réduit l'utilisation de la mémoire et la surcharge de calcul. Notre algorithme de chiffrement assure la non-répudiation et l'anonymat pour le système M2M ainsi que l'authentification et la gestion de confiance entre les appareils. L'efficacité et les performances de notre protocole sont évaluées par le vérificateur automatique de protocole cryptographique « ProVerif » et le simulateur « Cooja ». Les résultats obtenus ont montré que notre protocole est efficace contre les différents types d'attaques d'écoute et qu'il réduit la consommation d'énergie pour les objets connectés par rapport aux deux autres modèles proposés.

Dans nos travaux futurs, nous prévoyons prendre en charge les vulnérabilités liées aux objets connectés par rapport à la couche externe (réseaux Internet) de la maison intelligente. Ce travail est plus complexe parce qu'il prend en compte plusieurs scénarios d'accès par des entités externes. L'étude de ces scénarios sera suivie par des modifications au niveau du processus de contrôle et de notification afin d'adapter le protocole aux caractéristiques (puissance, mémoire ...) des objets connectés ainsi qu'aux exigences de la sécurité de l'information dans le contexte de la maison intelligente.

RÉFÉRENCES

[1] P. Suresh, J. V. Daniel, V. Parthasarathy, and R. 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)*, 2014: IEEE, pp. 1-8.

[2] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250-1258, 2017.

[3] M. Kranz, *Building the internet of things: Implement new business models, disrupt competitors, transform your industry*. John Wiley & Sons, 2016.

[4] I. T. Union, "Security framework for the Internet of things based on the gateway model," 2018.

[5] D. Prasad, N. N. Chiplunkar, and K. P. Nayak, "Performance comparison of IEEE 802.15. 4 and IEEE 802.15. 4e based MAC algorithm in wireless body sensor networks," in *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1119, no. 1: IOP Publishing, p. 012020.

[6] A. U. Echedom and I. F. Kakiri, "Prospects and challenges of internet of things application in library services," *Prospects*, 2021.

[7] D. Li, L. Deng, W. Liu, and Q. Su, "Improving communication precision of IoT through behavior-based learning in smart city environment," *Future generation computer systems*, vol. 108, pp. 512-520, 2020.

[8] S. Baskar, S. Periyanayagi, P. M. Shakeel, and V. S. Dhulipala, "An energy persistent range-dependent regulated transmission communication model for vehicular network applications," *Computer Networks*, vol. 152, pp. 144-153, 2019.

[9] S.-G. CRE. "Réseau intelligent (Smart Grid)." https://www.smartgrids-cre.fr/projets (accessed.

[10] N. C. S. R. Center. "Glossary terms and definitions." https://csrc.nist.gov/glossary/term/centralized_network (accessed.

[11] C. Dannen, *Introducing Ethereum and solidity*. Springer, 2017.

[12] F. Palmese, E. Longo, A. E. Redondi, and M. Cesana, "CoAP vs. MQTT-SN: Comparison and Performance Evaluation in Publish-Subscribe Environments," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021: IEEE, pp. 153-158.

[13] H. S. Zenalabdin, A. Buhari, and T. E. Nyamasvisva, "Performance analysis of iot protocol stack over dense and sparse mote network using cooja simulator," in *Journal of Physics: Conference Series*, 2020, vol. 1529, no. 5: IOP Publishing, p. 052007.

[14] K. Karimi and S. Krit, "Smart home-smartphone systems: Threats, security requirements and open research challenges," in *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, 2019: IEEE, pp. 1-5.

[15] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, "Poseidon: Mitigating interest flooding DDoS attacks in named data networking," in *38th annual IEEE conference on local computer networks*, 2013: IEEE, pp. 630-638.

[16] Y.-C. Wu, H.-R. Tseng, W. Yang, and R.-H. Jan, "DDoS detection and traceback with decision tree and grey relational analysis," in *2009 Third International Conference on Multimedia and Ubiquitous Engineering*, 2009: IEEE, pp. 306-314.

[17] M. S. Svetlin Nakov, Marina Shideroff, *Practical Cryptogtaphy for Developers* 2018.

[18] N. SULLIVAN. "A (relatively easy to understand) primer on elliptic curve cryptography." https://arstechnica.com/information-technology/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/ (accessed.

[19] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in internet of things: challenges and solutions," *arXiv preprint arXiv:1608.05187*, 2016.

[20] W. J. Buchanan. " Golang ECDH." https://asecuritysite.com/encryption/goecdh (accessed.

[21] Y. Waghid, Z. Waghid, and F. Waghid, "The fourth industrial revolution reconsidered: On advancing cosmopolitan education," *South African Journal of Higher Education*, vol. 33, no. 6, pp. 1-9, 2019.

[22] R. Ciffee, G. Sudha, S. Saranya, and G. K. Thyagesh, "Zigbee based automation systems for homes with the deployment of smart sensors," *Materials Today: Proceedings*, 2021.

[23] P. Gallo, K. Kosek-Szott, S. Szott, and I. Tinnirello, "SDN@ home: A method for controlling future wireless home networks," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 123-131, 2016.

[24] D. Han, H. Kim, and J. Jang, "Blockchain based smart door lock system," in *2017 International conference on information and communication technology convergence (ICTC)*, 2017: IEEE, pp. 1165-1167.

[25] P. Gallo, U. Q. Nguyen, G. Barone, and P. Van Hien, "DeCyMo: Decentralized cyber-physical system for monitoring and controlling industries and homes," in *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*, 2018: IEEE, pp. 1-4.

[26] Y. N. Aung and T. Tantidham, "Review of Ethereum: Smart home case study," in *2017 2nd International Conference on Information Technology (INCIT)*, 2017: IEEE, pp. 1-4.

[27] J. A. Stankovic, T. Le, A. Hendawi, and Y. Tian, "Hardware/software security patches for the internet of things," in *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2021: IEEE, pp. 240-245.

[28] M. I. M. Ariff, N. K. H. Kamaruzzaman, E. I. Zulkiflie, F. D. M. Fadzir, K. A. Salleh, and N. I. Arshad, "Design and Development for Smart Home via IoT Technology: A Work in Progress," in *2021 7th International Conference on Research and Innovation in Information Systems (ICRIIS)*, 2021: IEEE, pp. 1-4.

[29] R. Al-Syouf, B. Al-Duwairi, and A. S. Shatnawi, "Towards a Secure Web-Based Smart Homes," in *2021 12th International Conference on Information and Communication Systems (ICICS)*, 2021: IEEE, pp. 195-200.

[30] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2710-2723, 2013.

[31] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of attribute-based encryption: Toward data privacy in the IoT," in *2014 IEEE International Conference on Communications (ICC)*, 2014: IEEE, pp. 725-730.

[32] A. K. Hussain, "A modified RSA algorithm for security enhancement and redundant messages elimination using K-nearest neighbor algorithm," *IJISET-International Journal of Innovative Science, Engineering & Technology*, vol. 2, no. 1, pp. 858-862, 2015.

[33] S. Koppula and J. Muthukuru, "Secure digital signature scheme based on elliptic curves for internet of things," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 3, p. 1002, 2016.

[34] Z.-Q. Wu, Y.-W. Zhou, and J.-F. Ma, "A security transmission model for internet of things," *Jisuanji Xuebao(Chinese Journal of Computers)*, vol. 34, no. 8, pp. 1351-1364, 2011.

[35] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the Internet of Things," *Computers & Electrical Engineering*, vol. 37, no. 2, pp. 147-159, 2011.

[36] H. R. Hussen, G. A. Tizazu, M. Ting, T. Lee, Y. Choi, and K.-H. Kim, "SAKES: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6L0WPAN)," in *2013 Fifth international conference on ubiquitous and future networks (ICUFN)*, 2013: IEEE, pp. 246-251.

[37] I. Doh, J. Lim, S. Li, and K. Chae, "Key establishment and management for secure cellular machine-to-machine communication," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013: IEEE, pp. 579-584.

[38] A. Bartoli, J. Hernández-Serrano, O. León, A. Kountouris, and D. Barthel, "Energy-efficient physical layer packet authenticator for machine-to-machine networks," *Transactions on Emerging Telecommunications Technologies*, vol. 24, no. 4, pp. 401-412, 2013.

[39] H. Lamrani Alaoui, A. El Ghazi, M. Zbakh, A. Touhafi, and A. Braeken, "A highly efficient ECC-based authentication protocol for RFID," *Journal of Sensors*, vol. 2021, 2021.

[40] B. Blanchet, "Modeling and verifying security protocols with the applied pi calculus and ProVerif," *Foundations and Trends® in Privacy and Security*, vol. 1, no. 1-2, pp. 1-135, 2016.

[41] M. Martí, C. Garcia-Rubio, and C. Campo, "Performance evaluation of CoAP and MQTT_SN in an IoT environment," *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 31, no. 1, p. 49, 2019.

[42] L. Sitanayah, C. J. Sreenan, and S. Fedor, "A Cooja-based tool for coverage and lifetime evaluation in an in-building sensor network," *Journal of Sensor and Actuator Networks*, vol. 5, no. 1, p. 4, 2016.