Une approche légère basée BlockChain pour sécuriser les communications M2M dans l'IoT

Karam Eddine Bilami

a

UNIVERSITÉ DE STRASBOURG – UNIVERSITÉ HAUTE ALSACE

ECOLE DOCTORALE DE

MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR
ED 269

THÈSE DE DOCTORAT

Discipline : Informatique

Présentée par : BILAMI Karam Eddine

Le 24/06/2024

UNE APPROCHE LÉGÈRE BASÉE BLOCKCHAIN POUR SÉCURISER LES COMMUNICATIONS M2M DANS L'IOT

Devant le jury composé de :

Pr. Lyes KHOUKHI	ENSICAEN, Université de Normandie	Rapporteur
Pr. Jalel BEN OTHMAN	Université Paris-Saclay	Rapporteur
Dr. Sara BERRI	Cy Cergy Paris Université	Examinatrice
Pr. Jaime LLORET MAURI	Polytechnic University of Valencia	Examinateur
Pr. Pascal LORENZ	Université de Haute Alsace	Directeur de thèse
Dr. Jaafar GABER	Université de Technologie de Belfort-Montbéliard	Co-directeur de thèse

Remerciements

Je tiens à remercier Pr Pascal Lorenz pour avoir accepté de m'accueillir dans son équipe GRTC (Groupe de Recherche en Réseaux et Télécommunications) de l'IRIMAS (Institut de Recherches en Informatique, Mathématiques, Automatique et Signal) et de diriger cette thèse. Je lui suis reconnaissant pour son écoute et ses directives positives tout le long de la réalisation de ce travail.

Mes remerciements vont également à mon co-directeur de thèse, Dr Jaafar GABER pour son soutien fructueux et ses remarques constructives durant l'accomplissement de cette thèse.

J'exprime aussi mes sincères remerciements aux membres du jury :

- *Pr KHOUKHI et Pr BEN OTHMAN pour avoir évalué cette thèse en tant que rapporteurs. Je les remercie pour les commentaires et les suggestions qu'ils ont bien voulu tenir après lecture de ce manuscrit .*
- Pr. Jaime LLORET MAURI et Dr. Sara BERRI *d'avoir bien voulu examiner cette thèse et pour l'intérêt qu'ils ont porté à mon travail.*

Enfin, je voudrais remercier mes chers parents pour leurs encouragements permanents.

Table des matières

Remerciements 2
Table des matières 3
Liste des Figures 7
Liste des Tables 9
Glossaire 10
Résumé 12
Abstract 13

Chapitre 1 : INTRODUCTION GENERALE 14
1. Introduction 14
2. Contexte et Problématique 15
3. Objectif de la thèse 16
4. Contribution 16
5. Organisation de la thèse 17

Chapitre 2 : COMMUNICATION M2M (MACHINE A MACHINE) 19
1. Introduction 19
2. Définitions 19
3. Architecture M2M 20
3.1. Domaine réseau des dispositifs M2M 20
3.2. Domaine réseau de communication 21
3.3. Domaine d'Application 21
4. Normalisation des communications M2M 21
4.1. Introduction 21
4.2. Activités de normalisation de l'architecture ETSI 22
4.2.1. Normalisation de la couche application 22
4.2.2. Normalisation du domaine réseau de communication 23
4.2.3. Normalisation du domaine réseau des dispositifs M2M 23
4.3. OneM2M 24
4.4. LWM2M : Protocole de communication M2M Standardisé 25
5. Principales caractéristiques des communications M2M 26
6. Applications M2M 27
7. Plateformes M2M 29
8. Exigences des communications M2M 30
9. Défis posés par les communications M2M 30
9.1. Scalabilité des réseaux M2M 31
9.2. Communication M2M Pair à Pair 31
9.3. L'efficacité énergétique 32
9.4. La congestion dans les communications M2M 32
9.5. Qualité de Service (QoS) 32
9.6. Hétérogénéité 33

9.7. Autres défis..... 33
9.8. Sécurité des communications M2M :..... 34
10. Conclusion..... 34

Chapitre 3 : SÉCURITÉ DES COMMUNICATIONS M2M..... 35
1. Introduction..... 35
2. Services de sécurité pour les communications M2M..... 35
3. Vulnérabilités des Communications M2M..... 36
3.1. Au niveau du domaine réseau des dispositifs M2M 36
3.2. Au niveau du domaine réseau de communication..... 36
3.3. Au niveau du domaine application..... 37
4. Attaques contre les Communications M2M..... 37
4.1. Attaques physiques..... 37
4.2. Attaques logiques..... 38
4.2.1. Attaques Externes..... 38
4.2.2. Attaques Internes..... 39
4.3. Violation des données..... 39
5. Solutions de sécurité pour les communications M2M..... 42
5.1. Sécurité M2M au niveau Application..... 42
5.2. Solutions de sécurité du réseau des dispositifs M2M sans fil..... 43
5.2.1. Confidentialité..... 43
5.2.2. Gestion des clés de chiffrement..... 44
5.2.3. Protection de la Vie Privée (Privacy)..... 44
5.2.4. Contrôle d'accès..... 44
5.2.5. Intégrité..... 45
5.2.6. Authentification 45
5.2.6.1. Protocoles d'authentification traditionnels..... 46
5.2.6.2. Approches basées IA pour sécuriser les communications M2M dans l'IoT.. 47
5.2.6.3. Blockchain pour sécuriser les communications M2M dans l'IoT..... 48
6. Conclusion..... 49

Chapitre 4 : BLOCKCHAIN POUR LA SÉCURITÉ DES COMMUNICATIONS M2M..... 50
1. Introduction..... 50
2. Concept de la Blockchain..... 50
3. Les protocoles de consensus..... 52
3.1. Preuve de travail 52
3.2. Preuve d'Enjeu..... 53
3.3. Tolérance pratique aux fautes byzantines..... 53
3.4. Le consensus de Preuve d'Enjeu Déléguée..... 53
3.5. Raft..... 54
4. Types de la Blockchain..... 54
4.1. Blockchain Publique..... 54
4.2. Blockchain Privée..... 55
4.3. Consortium..... 55

5. Les plateformes Blockchain existantes....55
6. Evolution de la blockchain....57
6.1. Blockchain 1.0: Les cryptos-monnaies....58
6.2. Blockchain 2.0 : Les contrats intelligents....59
6.3. Blockchain 3.0 : La Convergence vers les DApp (applications décentralisées)....59
6.4. Blockchain 4.0 : L'intégration transparente avec l'industrie 4.0....60
7. Les avantages et les défis de la technologie Blockchain....60
7.1. Services offerts par la Blockchain....60
7.2. Les défis de la technologie Blockchain....61
1. Consommation d'énergie....61
2. Coût....61
3. Le statut réglementaire....61
4. Scalabilité....62
5. Attaques classiques sur une blockchain....63
8. Les Applications de la Blockchain....64
8.1. Les systèmes de vote basés sur la Blockchain....64
8.2. Améliorer la transparence des chaînes d'approvisionnements:....64
8.3. L'identité numérique....64
8.4. La blockchain pour M2M/ IoT....65
8 4.1. Blockchain pour la sécurité des communications M2M /IoT....65
9. Conclusion....66

Chapitre 5 : APPROCHES POUR LA SÉCURITÉ DES COMMUNICATIONS M2M
1. Introduction....68
2. Contributions....69
2.1. Mécanisme d'authentification léger basé sur la blockchain pour les communications M2M dans l'IoT (*Lightweight Blockchain-Based Scheme to Secure Wireless M2M Area Networks*)....69
2.1.1. Introduction....69
2.1.2. Architecture du domaine des dispositifs M2M....69
2.1.3. Schéma d'authentification proposé....71
2.1.3.1. Phase de pré-enregistrement....72
2.1.3.2. Phase d'enregistrement....75
2.1.3.3. Phase d'authentification....76
2.1.4. Évaluation des performances....77
2.1.3.1. Surcharge de communication....78
2.1.3.2. La surcharge de calcul....79
2.1.3.3. Latence moyenne....80
2.1.3.4. Consommation énergétique....81
2.1.5. Résultats de simulation....81
2.1.6. Analyse de sécurité....84
2.1.7. Discussion....85
2.1.8. Conclusion....86

2.2 Mécanisme d'authentification léger basé sur une blockchain à deux couches pour les communications M2M dans l'IoT (*Blockchain-based Authentication Protocol for Wireless M2M area Networks with Sidechain Integration*) 86
2.2.1. Introduction 87
2.2.2. Processus d'authentification 89
2.2.2.1. Phase d'enregistrement 89
2.2.2.2. Phase d'autorisation 89
2.2.2.3. Phase d'authentification 91
2.2.3. Évaluation des performances 92
2.2.3.1. Coût de stockage: 93
2.2.3.2. Analyse du Temps d'Authentification 93
2.2.4. Résultats de simulation 94
2.2.5. Analyse de sécurité 95
2.2.6. Conclusion 96
2.3. Protocole de communication multi-sauts sécurisé et économe en énergie (*SEEM-D2D: Secure and Energy Efficient Multi-hop D2D Communications in Wireless M2M Area Networks using Two-Layer Blockchain*) 97
2.3.1. Introduction 97
2.3.2. Modèle du système 98
2.3.2.1. Formation de clusters 99
2.3.2.2. Routage 100
2.3.3. Système de sécurité proposé 101
2.3.3.1. Phase d'enregistrement 102
2.3.3.2. Phase d'authentification 103
2.3.3.3. Phase de communication 105
2.3.3.4. Protocole de routage proposé 106
2.3.4. Évaluation des performances 112
2.3.4.1. Surcharge de calcul 113
2.3.4.2. Surcharge de communication 114
2.3.4.3. Consommation énergétique 115
2.3.5. Résultats de la simulation 115
2.3.6. Analyse de sécurité 120
2.3.6.1. Attaque de rejeu 120
2.3.6.2. Attaque de l'Homme du Milieu (MIM) 121
2.3.6.3. Attaque d'usurpation d'identité 121
2.3.6.4. Attaque par déni de service distribué (DDoS) 122
2.3.6.5. Authentification mutuelle 122
2.3.7. Discussion 123
2.3.8. Conclusion 123
Conclusion générale 125
Bibliographie 127
Liste des publications 135

Liste des Figures

Figure 2.1. Architecture M2M selon ETSI.

Figure 2.2. Architecture réseau pour la communication de type machine (MTC).

Figure 2.3. Architecture OneM2M : Vue fonctionnelle.

Figure 2.4. Architecture LWM2M.

Figure 2.5. Applications M2M.

Figure 4.1. Fonctionnement principal de la blockchain.

Figure 4.2. Le monde de la Blockchain : Un réseau distribué.

Figure 4.3. Le nombre de crypto-monnaies dans le monde.

Figure 4.4. Le trilemme de la blockchain.

Figure 5.1. Architecture de la couche M2M.

Figure 5.2. Station de base M2M.

Figure 5.3. Structure de bloc de la blockchain.

Figure 5.4. Phase de pré-enregistrement.

Figure 5.5. Contrat intelligent du schéma d'authentification.

Figure 5.6. Phase d’enregistrement.

Figure 5.7. Fonction GetAllTheInfo().

Figure 5.8. Phase d’authentification.

Figure 5.9. Comparaison des surcharges de communication.

Figure 5.10. Comparatif des coûts computationnels.

Figure 5.11. Latence moyenne, taux de livraison des paquets en fonction de la distance.

Figure 5.12. Consommation d'énergie.

Figure. 5.13. Architecture M2M basée sur la blockchain.

Figure. 5.14. Structure des blockchains.

Figure. 5.15. Diagramme du processus d'autorisation.

Figure. 5.16. Organigramme du processus d'authentification.

Figure. 5.17. Comparaison du temps d'authentification.

Figure. 5.18. Temps d'authentification vs nombre de nœuds.

Figure. 5.19. Objectifs et propriétés de sécurité.

Figure. 5.20. Les principales spécifications.

Figure. 5.21. Résultats de la vérification du protocole.

Figure. 5.22. Architecture M2M.

Figure. 5.23. Diagramme de séquence du protocole proposé.

Figure. 5.24. Structure des blocs.
Figure. 5.25. Contrat intelligent de la chaîne principale.
Figure. 5.26. Fonction Générer MAC.
Figure. 5.27. Pseudocode de la phase d'authentification.
Figure. 5.28. Fonction de la sidechain : Authenticate Device.
Figure. 5.29. Le message “Route Request”.
Figure. 5.30. Le message “Route Reply”.
Figure. 5.31. Diagramme de flux du processus de routage.
Figure. 5.32. Erreur d’itinéraire.
Figure. 5.33. Schéma de communication proposé..
Figure. 5.34. La table de routage du dispositif N3.
Figure. 5.35. Comparaison de la surcharge de calcul.
Figure.5.36. Consommation d'énergie par type de communication à différentes distances.
Figure. 5.37. Énergie consommée par l'établissement de la route vs nombre de sauts.
Figure. 5.38. Consommation d'énergie de l'établissement de route avec et sans sécurité.
Figure. 5.39. Temps d'authentification en fonction du nombre de nœuds.
Figure. 5.40. Objectifs et propriétés de sécurité.
Figure. 5.41. Les principales spécifications.
Figure. 5.42. Résultats de vérification du protocole.

Liste des Tables

Table 2.1: Attaques sur les réseaux M2M.

Table 4.1 :Comparaison des principaux mécanismes de consensus.

Table 4.2 :Comparaisons entre la blockchain publique, consortium et privée.

Table 4.3 :Comparaison des plateformes de contrats intelligents.

Table 5.1 : Notations et longueurs des paramètres.

Table 5.2 : Coût de communication.

Table 5.3 : Le temps approximatif pour chaque opération cryptographique.

Table 5.4 : Comparaison du coût de calcul (CC) de l'authentification en secondes.

Table 5.5 : Paramètres de simulation.

Table 5.6 : Comparaison avec d'autres protocoles.

Table 5.7 : Le coût de la surcharge de calcul du routage.

Table 5.8 : Coût de surcharge de communication (CSC) pour N sauts.

Table 5.9 : Paramètres de simulation.

Table 5.10 : Classement par type de communication à des distances variables.

Table 5.11 : Caractéristiques de certains protocoles de routage M2M existants.

Glossaire

- 3GPP 3rd Generation Partnership Project
- 6LoWPAN IPv6 over Low power Wireless Personal Area Networks
- AMQP Advanced Message Queuing Protocol
- API Application Programming Interface
- CDMA Code Division Multiple Access
- D2D Device to Device
- D2I Device To infrastructure
- I2D Infrastructure To Devic
- DDoS Distributed Denial of Service
- DoS Denial of Service
- ECC Elliptic Curve Cryptography
- ECDH Elliptic Curve Diffie-Hellman
- EDGE Enhanced Data Rates for Global Evolution
- ETSI European Telecommunications Standards Institute
- eNB Evolved Node Base station
- gNB next Generation Node Base station
- GPRS General Packet Radio Service
- GSM Global System for Mobile Communications
- H2H Human-to-Human
- H2M Human-to-Machine
- IDS Intrusion Detection System
- IPS Intrusion Detection System
- IEEE Institute of Electrical and Electronics Engineers
- IETF Internet Engineering Task Force
- IIOT Industrial IoT
- IoT Internet-of-Things
- IP Internet Protocol
- IPSec IP security
- ISDN International Subscriber Directory Number
- LEACH Low-Energy Adaptive Clustering Hierarchy
- LLN Low Power and Lossy Network
- LTE Long Term Evolution
- LTE-A LTE Advanced
- M2M Machine to Machine
- MAC Message Authentication Code
- MAC Medium Access Control
- MEC Mobile Edge Computing
- MIM Man In The Middle Attack
- MTC Machine Type Communication
- mMTC massive Machine Type Communication
- MSISDN Mobile Station International Subscriber Directory Number
- MTCD Machine Type Communication Device

- MTCG Machine Type Communication Gateway
- MQTT Message Queuing Telemetry Transport
- NS Network Simulator
- P2P Peer To Peer
- PBFT Practical Byzantine Fault Tolerance
- PDR Packet Delivery Ratio
- PoS Proof of Stake
- PoW Proof of Work
- RCSF Réseau de Capteurs Sans Fil
- RFID Radio Frequency Identification
- RSSI Received Signal Strength Indicator
- RSA Rivest, Shamir und Adleman
- SCADA Supervisory Control and Data Acquisition
- SPF Single Point of Failure
- SSL Secure Socket Layer
- TLS Transport Layer Security
- TDMA Time Division Multiple Access
- TPM Trusted Platform Module
- UMTS Universal Mobile Telecommunications System
- VPN Virtual Private Network
- WiFi Wireless Fidelity
- WiMAX Worldwide Interoperability for Microwave Access
- WPAN Wireless Personal Area Networks
- WSN Wireless Sensor Network
- XMPP Extensible Messaging and Presence Protocol

Résumé

Le concept M2M (ou Machine à Machine) fait partie des technologies de l'Internet des objets (IdO) ou l'IoT (Internet of Things), qui permettent à des machines ou à des objets intelligents hétérogènes équipés de capteurs intégrés et interconnectés par des réseaux, de détecter, de prélever, de traiter et de communiquer des informations sans intervention humaine.
La communication M2M revêt une grande importance dans divers domaines tels que l'industrie, la santé, l'agriculture, la logistique, la domotique et les villes intelligentes, où elle contribue à l'automatisation et à l'optimisation des opérations.
En raison du déploiement à grande échelle des dispositifs M2M dans différents lieux non protégés, et en raison de leur fonctionnement autonome, sans intervention humaine, les communications M2M sont vulnérables à différentes attaques logiques et physiques.
La sécurisation de ces communications représente un défi majeur. Cependant, les protocoles de sécurité classiques, appliqués dans l'Internet, sont généralement inappropriés pour la protection des dispositifs M2M, en raison des ressources limitées de ces machines en termes d'énergie, de stockage et de traitement. De nouvelles approches sont donc nécessaires pour définir comment assurer la sécurité de ces communications.
Comme solution alternative, la technologie blockchain, qui fait l'objet d'une attention croissante en tant que système distribué, est capable de répondre aux préoccupations de sécurité des communications M2M. Elle permet d'enregistrer toutes les transactions dans un registre distribué (Distributed Ledger), rendant toute falsification pratiquement impossible. En théorie, la blockchain peut fournir des services de sécurité tels que l'intégrité, l'authentification, la disponibilité et l'autorisation d'accès; cependant, il y a des défis à surmonter lorsqu'elle est utilisée dans des contextes M2M et IoT en raison d'une évolutivité et d'une mise à l'échelle (scalabilité) contraignante, et en raison des limitations des ressources des appareils utilisés.
A travers cette thèse de recherche, nous avons étudié la capacité de la technologie blockchain à répondre aux problèmes de sécurité introduits par les communications M2M dans l'IoT. Dans cette optique, nous avons examiné les défis et les problèmes de sécurité présentés dans la littérature pour les communications M2M. Nous avons, dans un premier temps, proposé un protocole d'authentification léger basé sur une blockchain privée, conçu pour sécuriser les réseaux des dispositifs M2M sans fil.
Ensuite, et pour répondre au souci de la scalabilité des communications M2M, nous avons introduit un mécanisme d'authentification utilisant une blockchain à deux couches (une chaîne principale appelée Mainchain et une chaîne secondaire ou latérale appelée Sidechain).
Finalement, un protocole de communication multi-sauts sécurisé et économe en énergie nommé SEEM-D2D, utilisant l'architecture blockchain à deux couches, est proposé pour sécuriser des communications de type D2D dans les réseaux M2M.
Ces contributions ont été évaluées et comparées à des solutions présentées dans la littérature, démontrant de meilleures performances en termes de surplus (overhead) introduits en espace stockage, en temps d'exécution et en consommation énergétique.
L'évaluation a été effectuée par simulation en utilisant NS3. Une analyse de la solution de sécurité proposée a été conduite en utilisant l'outil de vérification AVISPA.

Mots-clés : M2M, IoT, Blockchain, Sécurité, Routage, D2D, Authentification, Scalabilité, Efficacité énergétique.

Abstract

The M2M (or Machine-to-Machine) concept is part of today's Internet of Things (IoT) technologies, which enable heterogeneous and intelligent objects (or machines) equipped with integrated sensors and interconnected through networks, to detect, retrieve, process and communicate information without human intervention.
M2M communication is of vital importance in fields such as industry, healthcare, agriculture, logistics, home automation and smart cities, where it contributes to the automation and optimization of different operations.
Due to the large-scale deployment of M2M devices in various unprotected locations, and their autonomous operation without human intervention, M2M communications are vulnerable to a variety of logical and physical attacks.
Securing these communications represents a major challenge. However, conventional security protocols applied in the Internet are generally unsuitable for protecting M2M devices, due to the limited resources of these machines, in terms of energy, storage and processing. New approaches are therefore needed to define how to ensure the security of these communications.
As an alternative, blockchain technology, which is receiving increasing attention as a distributed system, is capable of addressing the security concerns of M2M communications. It enables all transactions to be recorded in a Distributed Ledger, making any data alteration practically impossible. In theory, blockchain can provide security services such as integrity, authentication and access authorization. However, there are challenges to overcome when used in the context of M2M/IoT, due to constraining scalability and resource limitations of the M2M devices.
Through this thesis, we investigated the ability of blockchain technology to address the security issues introduced by M2M communications in the IoT. To this end, we studied the security challenges and issues presented in the literature for M2M communications. We first proposed a lightweight authentication protocol based on a private blockchain, designed to secure wireless M2M device networks.
In addition, and to address the scalability and energy efficiency concerns of M2M communications, an authentication mechanism using a two-layer blockchain (mainchain and sidechain) is proposed. Finally, a secure and energy-efficient multi-hop communication protocol, named ‘SEEM-D2D’ is designed, using two-layer blockchain architecture in order to enhance security and energy efficiency through D2D communications in M2M networks.
These contributions have been evaluated and compared with existing solutions, demonstrating better performance in terms of storage overhead, execution time and energy, The evaluation has been carried out by simulation using NS3. An analysis of the proposed security solutions has been conducted using AVISPA verification tools.

Keywords : M2M, IoT, Blockchain, Security, Routing, D2D, Authentication, Scalability, Energy efficiency.

Chapitre 1

INTRODUCTION GENERALE

1. Introduction

L'erreur est humaine ! Oui, c'est vrai, mais l'erreur humaine est derrière la majorité des accidents et des divulgations de secrets dans beaucoup de domaines (industriel, médical, informatique, etc.). Dans le domaine des TIC, il est reconnu que le facteur humain représente une source de failles de sécurité, qui sont exploitées par les cybercriminels pour effectuer des accès non autorisés, voler des informations d'identification, infecter les systèmes, etc. Éliminer des sources d'erreurs humaines, dont les causes sont en général la fatigue, le sommeil, l'oubli, la distraction, et la complaisance, consiste à remplacer l'homme par la machine, et à adopter le paradigme M2M (Machine to Machine) dans l'accomplissement de certaines fonctions qui peuvent être automatisées dans différents domaines.

En outre, en plus de permettre de réduire la probabilité d'accident et d'erreurs non intentionnelles, les communications M2M présentent l'avantage de réduire les temps d'exécution induits par l'intervention humaine.

Les communications de machine à machine (M2M) est un concept qui concerne l'automatisation des échanges de données entre deux ou plusieurs machines, et l'exécution de tâches qui ne nécessitent pas une intervention humaine directe [1].

Historiquement, l'échange d'informations entre machines date du début du 20^{e} siècle. Jusqu'à la fin des années 1960, les communications de machine à machine étaient utilisées dans des applications de télémétrie pour le contrôle et la gestion d'infrastructures électriques, de gaz et d'eau par le biais d'une connexion exclusivement filaire et une communication directe entre les appareils.

L'adoption à grande échelle des communications de machine à machine, appelées aussi communications de type machine (MTC), a commencé dans les années 1980 avec la prolifération des systèmes SCADA (Supervisory Control and Data Acquisition), déployés dans les usines et dans les systèmes de sécurité des entreprises. Les systèmes SCADA sont considérés comme les premiers systèmes M2M [2]

Plus récemment, les communications sans fil ont donné une autre dimension aux applications M2M, qui ont vu un regain d'intérêt, en permettant une connectivité omniprésente entre des dispositifs MTC et un serveur M2M, ou une connectivité entre deux dispositifs MTC. L'échange de données entre les différents dispositifs connectés se fait sans intervention humaine, en utilisant différentes technologies de communication sans fil (802.11, 802.15.1, 802.15.4, LTE-4G, 5G et au-delà …).

La communication M2M sans fil peut être définie comme un concept, faisant partie des technologies de l'Internet des objets (IoT), qui prend en charge des dispositifs intelligents hétérogènes interconnectés par des réseaux, pour la détection, le traitement et la communication d'informations, sans intervention humaine.

Communément, Il s'agit d'un autre type de communications dans l'IoT, en plus des communications H2H (Human to Human) et H2M (Human to Machine). La différence réside dans le fait que le M2M peut utiliser le mode P2P (Peer to Peer) ou la communication directe en plus des communications via l'internet, tandis que le H2M et le H2H connectent des appareils à d'autres appareils ou à des personnes uniquement via l'internet.

Les applications M2M sans fil (par exemple dans les villes et maisons intelligentes, la santé à distance, les transports intelligents, etc.) exploitent les données générées par une grande variété de dispositifs (devices en anglais, que nous désignons aussi, tout au long de ce manuscrit, par appareils ou encore machines) hétérogènes, en temps réel ou après stockage en vue d'un traitement ultérieur.
De nos jours, en raison des caractéristiques avantageuses de la télémétrie sans fil par rapport à la télémétrie classique : transmission sur de plus longues distances, déploiement plus facile et plus étendu à faible coût, elle est largement utilisée par plusieurs applications, notamment dans la télémédecine ou la-santé à distance à travers des capteurs de tension artérielle, de glycémie, etc.
Cependant, les communications M2M sont confrontées à un certain nombre de défis à cause des contraintes imposées par les dispositifs ou appareils terminaux, qui sont en général des systèmes RFID, des capteurs sans fil, des actionneurs, etc. disposant de ressources limitées en énergie, et en capacité de traitement et de stockage. En outre, en raison du déploiement à grande échelle de ces dispositifs dans différents lieux non protégés (cités, hangars, entreprises, forêts, montagnes, véhicules, chaînes d'approvisionnement...), ces dispositifs se trouvent confrontés à plusieurs problèmes tels que l'extensibilité ou la mise à l'échelle, l'hétérogénéité des systèmes, les technologies de communication, et surtout la sécurité des dispositifs M2M qui sont censés fonctionner sans protection physique et sans intervention humaine.

2. Contexte et Problématique

De nombreuses applications M2M dans les différents domaines (militaire, santé, maison intelligente, industriel, etc.) traitent des informations confidentielles et vitales, nécessitant sûreté et confidentialité, étant donné que les résultats et les décisions qui s'ensuivent sont prises sur la base des données générées par ces dispositifs. La réussite de ces applications reste tributaire des solutions de sécurité qu'elles implémentent.

Dans la littérature, on retrouve plusieurs études synthétisant diverses solutions de sécurité pour contrer les vulnérabilités des communications M2M à différents niveaux (interfaces physiques, protocoles de liaison, de réseau et d'application). Ces solutions utilisent généralement les environnements réseaux intégrant les technologies Cloud, Fog ou Edge Computing, pour héberger les serveurs et les traitements assurant les différents services de sécurité basés sur de lourds crypto-systèmes cryptographiques. [1][3][4][5]

Toutes ces études conviennent que les protocoles de sécurité classiques, utilisés pour les applications Internet, sont généralement inappropriés pour la protection des dispositifs M2M/IoT, en raison des ressources limitées de ces machines qui ne peuvent pas supporter les

lourdes opérations cryptographiques utilisées par ces protocoles. En plus, les approches fondées sur un contrôle centralisé présentent l'inconvénient du SPF (Single Point of Failure). Peu d'études dans la littérature traitent le sujet en proposant des solutions décentralisées, légères('lightweight'), qui tiennent compte des contraintes et des limites des dispositifs terminaux ou des capteurs sans fil utilisés dans les applications M2M.

Comme solutions alternatives, de nouvelles approches décentralisées ont été récemment proposées pour sécuriser les communications M2M. La technologie blockchain, faisant l'objet d'une attention croissante, en fait partie en tant que système distribué capable de répondre aux préoccupations de sécurité des communications M2M [6][7][8][9].

3. Objectif de la thèse

L'objectif principal de cette thèse est de définir un modèle de communications M2M sécurisées, basé sur des solutions légères de sécurité, peu consommatrices d'énergie, distribuées, largement déployées en permettant la communication entre les appareils (ou machines) de différents réseaux du domaine des dispositifs M2M.

La blockchain permet d'enregistrer toutes les transactions dans une base de données partagée ou un registre distribué (Distributed Ledger). Toute modification peut être facilement détectée. En théorie, la blockchain peut fournir des services de sécurité tels que l'intégrité, l'authentification et l'autorisation d'accès. Cependant, il y a des défis à surmonter lorsqu'elle est utilisée dans des contextes M2M/IoT, en raison des difficultés d'extension ou de mise à l'échelle (scalabilité), et des contraintes imposées par les ressources limitées des appareils utilisés [6].

À travers cette thèse, il s'agit d'étudier et d'évaluer la capacité de la technologie blockchain à répondre aux problèmes de sécurité dans les communications M2M. Dans cet objectif, nous devons d'abord examiner les contraintes et les défis de sécurité posés par les communications M2M.

Le but final étant de proposer une architecture de sécurité légère et appropriée pour les communications M2M, avec de bonnes performances en termes de stockage, de temps d'exécution et d'énergie comparativement aux approches de sécurité traditionnelles.

4. Contribution

Dans l'optique de contribuer dans le domaine de la sécurité des communications M2M, nous avons défini dans un premier temps un protocole léger basé blockchain pour l'authentification des communications M2M. Étant donné que les appareils limités en ressources ne sont pas en mesure de supporter le traitement lourd que la technologie blockchain introduit pour les opérations minières et cryptographiques, une blockchain privée a été adoptée pour être placée sur un ensemble de stations de base au niveau 'Edge', c'est-à-dire proche des champs de détection ou de la zone des dispositifs M2M, ce qui présente l'avantage des délais courts et un contrôle précoce des appareils M2M. La blockchain est conçue pour contenir toutes les informations relatives à chaque appareil.

En extension à ce travail, nous avons conduit des recherches pour la sécurisation des communications M2M grâce à l'introduction d'une blockchain à deux niveaux (une chaîne principale et une chaîne secondaire). La blockchain principale est utilisée pour garantir la sécurité en termes d'authentification, de confidentialité et de disponibilité, alors que la chaîne secondaire est utilisée pour assurer la scalabilité (ou l'évolutivité) des réseaux de dispositifs M2M.
Dans un deuxième temps, un modèle de communication multi-sauts sécurisé qui tient compte de la consommation en énergie, a été proposé pour permettre un échange de données sécurisé entre les dispositifs M2M dans l'un des deux modes suivants :

- Avec infrastructure sous le contrôle des stations de base
- Sans infrastructure, en utilisant les communications D2D (Device to Device) en cas de défaillance de la station de base.

Les solutions proposées pour assurer l'authentification des communications dans le domaine des dispositifs M2M, ainsi que le protocole de routage sécurisé D2D multi-sauts dans ce domaine, ont été évalués en termes de stockage, de temps d'exécution et de consommation d'énergie, avec des mesures de performances comparées à d'autres approches de sécurité présentées dans la littérature. L'évaluation a été effectuée par simulation en utilisant NS3. Une analyse des solutions de sécurité proposées a été conduite en utilisant l'outil de vérification AVISPA [10].

5. Organisation de la thèse

Ce manuscrit est organisé en plus d'une introduction générale, de quatre chapitres, dont les trois premiers présentent des états de l'art relatifs aux domaines d'intérêt investis dans le cadre de notre recherche, à savoir: le concept M2M, la technologie Blockchain, la sécurité des communications M2M. Le dernier chapitre est consacré à nos contributions.
Le manuscrit débute avec un premier chapitre introductif au domaine de recherche, présentant l'intérêt du sujet, l'objectif et la problématique de cette thèse, ainsi que les contributions réalisées dans le cadre de ce projet de thèse.
Le deuxième chapitre présente des généralités sur les communications M2M, abordant les définitions, les caractéristiques, les applications, ainsi que les architectures et les spécifications de normalisation des communications M2M. Enfin, les différents défis auxquels ces communications font face en matière d'efficacité énergétique, d'hétérogénéité, de scalabilité, de QoS, et de sécurité y sont discutés.
Le troisième chapitre est entièrement consacré à la sécurité des communications M2M, commençant par les exigences et les services de sécurité que doivent assurer les systèmes et plateformes M2M. Le chapitre s'étale longuement sur les vulnérabilités et les attaques contre les réseaux M2M.
Le chapitre 4 décrit la technologie blockchain et son application pour la sécurité des communications M2M. Après une introduction présentant les concepts de base de la blockchain et les protocoles de consensus associés, il explore les différents types de blockchain ainsi que leur évolution. Les avantages et les défis de la technologie blockchain sont examinés en mettant en exergue les attaques classiques qui peuvent affecter ces

systèmes. Enfin, le chapitre présente des applications pratiques de la blockchain dans les réseaux M2M/IoT.

Le chapitre cinq de cette thèse révèle nos contributions dans le domaine de la sécurisation des communications M2M. Le premier apport de notre recherche réside dans la conception d'un protocole d'authentification basé sur la technologie blockchain, conçu spécifiquement pour sécuriser les réseaux des dispositifs M2M sans fil. Notre deuxième contribution consiste en un protocole d'authentification pour les communications M2M sans fil basé sur deux blockchain (avec intégration d'une sidechain). Enfin, un protocole de communication multi-sauts, nommé SEEM-D2D, représentant un troisième apport dans cette thèse, est proposé dont le but est de garantir la sécurité des réseaux des dispositifs M2M en utilisant des communications D2D sécurisées et efficaces en énergie.

Enfin, une conclusion générale vient parachever ce manuscrit, en récapitulant les travaux réalisés, discutant les points forts des systèmes de sécurité proposés ainsi que leurs insuffisances avec des perspectives pour des travaux futurs.

Chapitre 2

COMMUNICATION M2M (MACHINE A MACHINE)

1. Introduction

Machine à Machine, ou M2M (Machine to Machine) est un concept général qui est utilisé pour décrire toute technologie permettant à des dispositifs (ou machines), connectés en réseau, d'échanger des informations et d'effectuer des actions sans l'assistance manuelle de l'homme [11].
En fait, le concept M2M n'est pas récent; la technologie M2M a été d'abord adoptée à la fin des années 1970, dans les secteurs de la fabrication et de l'industrie, avec des systèmes tels que le SCADA (Supervisory Control and Data Acquisition ou système de supervision industrielle), qui traite en temps réel un grand nombre de mesures de phénomènes physiques (pression, température…) et de contrôle à distance des installations [12].
Les progrès de la technologie sans fil M2M ont permis, à partir des années 2000, de mettre en œuvre des solutions sans fil et de remplacer les connexions câblées entre les capteurs et les RTU (remote terminal units) et PLC (programmable logic controller) par des liaisons sans fil. Ces dernières ont permis également de placer des capteurs et des unités distantes dans des zones d'exploitation qui étaient auparavant difficilement (ou non) accessibles par le câblage, etc.
Le M2M, notamment dans sa version sans fil, représente un sujet de grand intérêt pour l'avenir de l'informatique et des réseaux de communication. Il s'agit d'une révolution technologique qui a un impact profond sur notre vie quotidienne. Le M2M a permis la définition de plus en plus d'applications dans plusieurs secteurs, que nous présentons dans ce chapitre.

2. Définitions

On trouve plusieurs définitions, plus ou moins proches, des communications M2M dans la littérature, les plus pertinentes sont présentées ci-après :
Définition 1 : Le concept M2M (Machine-to-Machine) représente un ensemble de technologies permettant l'échange de données entre machines connectées sur un réseau de communication câblé ou sans fil ayant pour objectif l'exécution d'opérations sans intervention humaine. [13] [14]

Définition 2 : Les communications M2M sans fil, en particulier, font référence à un concept qui permet l'échange de données entre différents dispositifs connectés sans intervention humaine, en utilisant différentes technologies de communication sans fil telles que (802.11, 802.15.1, 802.15.4, LTE-4G, 5G et au-delà, etc.). [15]

De manière générale, la définition suivante fait l'objet d'un large consensus :

Définition 3: les communications M2M désignent toute technologie qui permet l'échange de données entre divers dispositifs (smartphones, dispositifs WPAN de santé, capteurs, contrôleurs intégrés, actionneurs,...) sur des réseaux câblés ou sans fil, tels que GSM/GPRS/EDGE, CDMA, RFID, Wi-Fi, ZigBee, WiMAX, xDSL, afin d'effectuer des actions sans intervention humaine. [16][17]

Vu la grande variété des applications, le nombre d'appareils connectés ne cesse de croître. Selon Statistica, un portail en ligne spécialisé en statistiques issues de données d'instituts, d'études de marché et d'opinion ainsi que de données provenant du secteur économique, le nombre d'appareils de l'Internet des objets (IoT) dans le monde devrait presque doubler, passant de 15,1 milliards en 2020, à plus de 29 milliards d'appareils IoT en 2030 [18].

La prochaine évolution des communications M2M s'appuiera sur les réseaux 5G. On s'attend à l'avenir proche à l'apparition de la "communication massive de type machine" (mMTC). Il s'agit d'un avenir qui prévoit l'installation à très grande échelle (de l'ordre du million d'appareils par kilomètre) de réseaux sans fil à faible puissance pour les applications M2M. Pour répondre aux besoins d'applications exigeantes, une technologie mMTC nécessite également une latence de communication d'une milliseconde entre les nœuds et une très grande fiabilité radio.

3. Architecture M2M

Une architecture générique de bout en bout d'un système M2M se compose de trois domaines interconnectés, telle qu'elle est définie par l'Institut européen des normes de télécommunication ETSI [19] et illustrée dans la Figure 2.1.

1. Domaine du dispositif M2M.
2. Domaine du réseau.
3. Domaine de l'application.

3.1. Domaine réseau des dispositifs M2M.

Dans ce domaine, un réseau de dispositifs M2M peut être formé par un grand nombre d'appareils (tels que des capteurs, des actionneurs et des compteurs intelligents) et de passerelles (points de collecte de données/concentrateurs). Ces dispositifs recueillent des données de perception et de surveillance du domaine des dispositifs M2M pour les transmettre à une passerelle [2].Celle-ci assure l'interconnexion des dispositifs M2M au domaine réseau de communication. S'il existe plusieurs passerelles dans un domaine réseau M2M, elles peuvent en outre communiquer directement entre elles (communication pair à pair) pour prendre des décisions communes. Les dispositifs déployés dans un domaine M2M peuvent varier selon le type d'application.

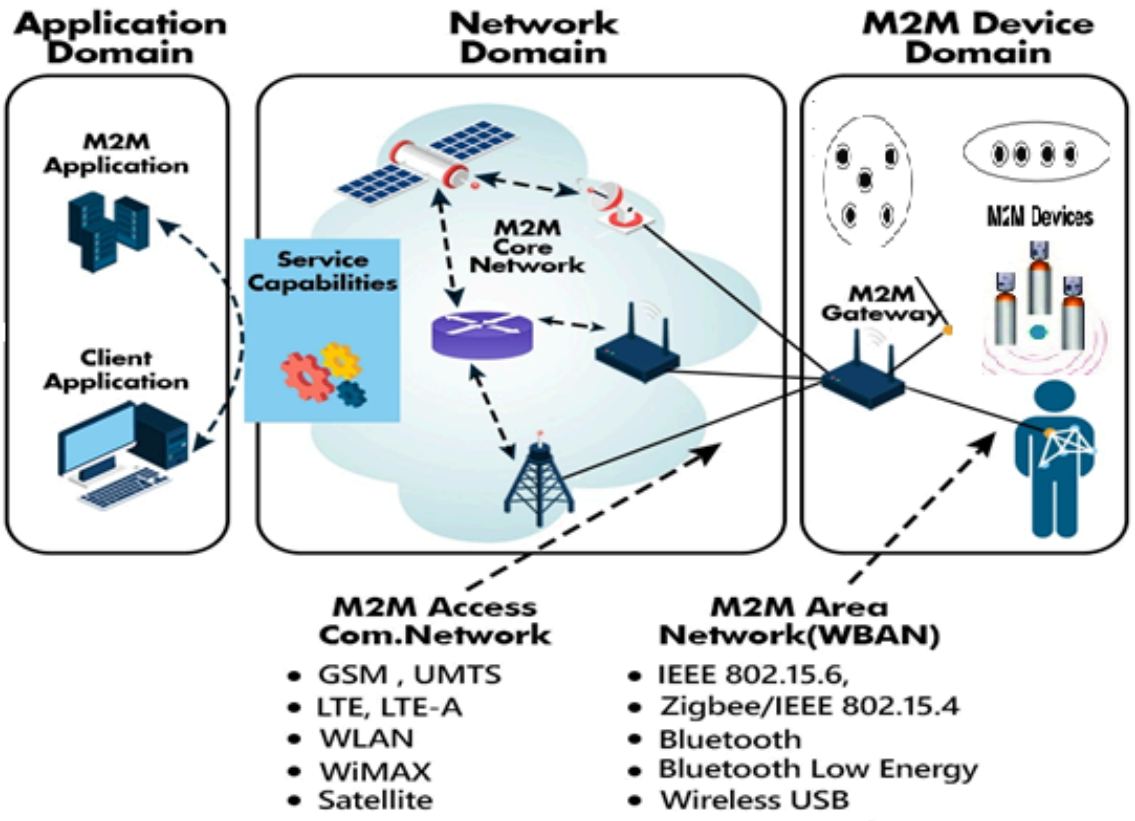


Figure 2.1. Architecture M2M selon ETSI [19]

3.2. Domaine réseau de communication

Le domaine réseau de communication sert d'interface entre le domaine des dispositifs M2M et le domaine Application M2M. Dans ce domaine, les données sont transmises sous forme de paquets aux serveurs du domaine Application, par un chemin passant par un canal à saut unique ou à sauts multiples. Les protocoles de réseaux filaires/sans fil à longue portée tels que les réseaux satellitaires, téléphoniques DSL, WiMAX et les réseaux cellulaires 3G/4G et 5G sont utilisés pour fournir une couverture étendue et fiable [20].

3.3. Domaine d'Application

Dans ce domaine, on trouve les serveurs qui supportent des applications qui permettent à des utilisateurs autorisés de bénéficier de la télédétection, du contrôle à distance et de la collecte de données à distance [21].

4. Normalisation des communications M2M

4.1. Introduction

La normalisation M2M joue un rôle essentiel dans le développement des communications M2M, en permettant d'assurer l'uniformité et l'interopérabilité. Pour ce faire, plusieurs organismes et des alliances de normalisation (3GPP, ETSI, IETF et oneM2M) qui sont axées

sur la technologie de l'information et ses applications, ont convenu d'élaborer conjointement une norme mondiale pour les communications M2M, aboutissant à une architecture commune du système M2M [22]. Chaque partie prenante exécute une ou plusieurs tâches spécifiques. Ces tâches peuvent comprendre la connectivité, l'interopérabilité, le déploiement, l'activation, les services d'intégration, etc. [23]

L'ETSI s'intéresse à l'architecture des services M2M, à ses composants et aux interactions entre les trois domaines (domaine réseau des dispositifs M2M, le domaine réseau de communication et le domaine des applications M2M). L'architecture M2M de l'ETSI est l'architecture de référence actuelle pour les communications M2M de bout en bout.

En revanche, les activités du 3GPP se sont concentrées sur les communications M2M qui peuvent être prises en charge par les réseaux cellulaires mobiles. Le 3GPP a choisi le nom de communication de type machine (MTC) pour désigner la communication M2M. Des travaux de normalisation ont été menés pour le domaine réseau des dispositifs M2M, afin d'optimiser l'accès à l'infrastructure du domaine du réseau de communication pour garantir efficacement les services M2M.

Le 3GPP a défini des services de transport et de communication pour faciliter les communications entre l'équipement utilisateur MTC et les applications MTC de bout en bout [24]. La Figure 2.2 illustre le modèle de référence architectural fourni par le 3GPP pour le MTC.

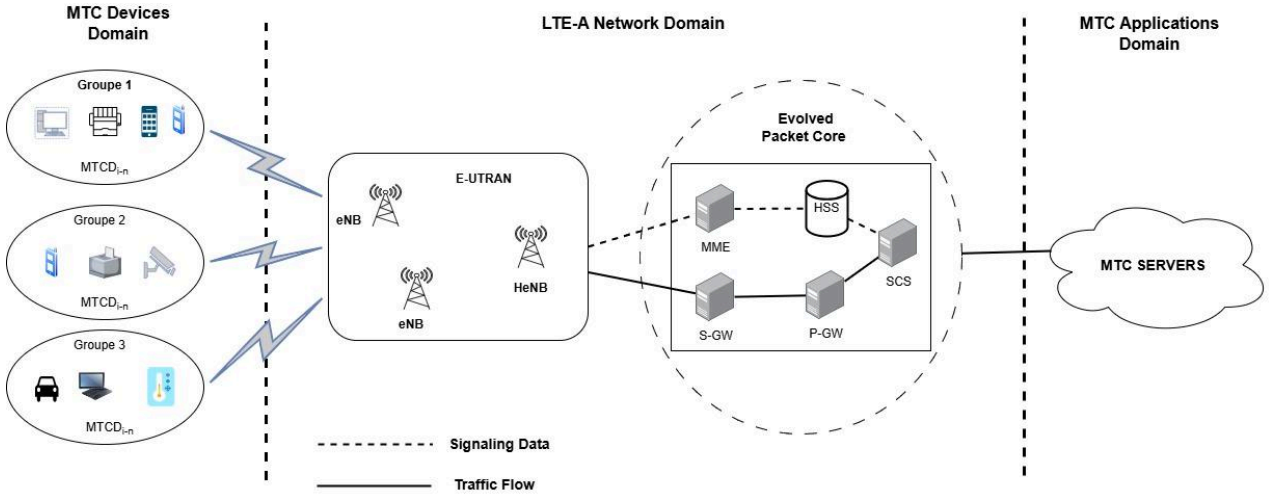


Figure 2.2. Architecture réseau pour la communication de type machine (MTC). [24]

4.2. Activités de normalisation de l'architecture ETSI

Le champ d'application des divers organismes de normalisation actifs, relatif au système M2M et son architecture, s'étend des couches du réseau des dispositifs M2M et du domaine réseau de communication, à la normalisation de la couche d'application.

4.2.1. Normalisation de la couche application

Le champ d'action de l'Institut européen des normes de télécommunication s'étend à la normalisation de la couche d'application qui est indépendante du réseau de communication

sous-jacent. L'objectif de l'ETSI à ce niveau, comprend les spécifications des cas d'utilisation, les exigences du service M2M, l'architecture fonctionnelle et la normalisation de l'interface.
Le groupe de travail de l'IETF sur l'environnement RESTful contraint (CoRE) a défini le protocole d'application contraint (CoAP Constrained Application Protocol) [25], qui fournit un protocole d'application pour manipuler les ressources des réseaux de dispositifs M2M. Pour réduire le chevauchement des normes et éviter la création de normes M2M concurrentes, huit organismes de normalisation dont l'ETSI, ont mis en place la normalisation oneM2M [26].

4.2.2. Normalisation du domaine réseau de communication

L'objectif de la norme 3GPP à ce niveau est d'intégrer le M2M ou Machine Type Communication (MTC) dans le réseau cellulaire 3GPP existant. Les organismes de normalisation du domaine réseau de communication, notamment le 3GPP, l'IEEE 802.16p [27] et le forum WiMAX [28]. L'objectif est de spécifier les exigences M2M au niveau de l'architecture du système de réseau, les utilisations, les modèles de déploiement basés sur les protocoles IEEE 802.16, afin de permettre une série d'applications M2M dans lesquelles les communications des appareils couvrent une large zone sans fil avec une interface radio avec une faible consommation d'énergie.
Les principaux éléments de normalisation sont les identificateurs, l'adressage, le déclenchement des appareils, le contrôle de l'encombrement et de la surcharge du système.

4.2.3. Normalisation du domaine réseau des dispositifs M2M

L'IEEE gère le groupe de travail 802.15.4 [29], dont l'objectif est la normalisation de la couche PHY/MAC des réseaux personnels sans fil (WPAN), qui se composent d'appareils à faible consommation. La norme 802.15.4 est adoptée comme couche PHY/MAC de ZigBee [30].
L'Internet Engineering Task Force (IETF) a organisé un groupe pour travailler sur le 6LoWPAN [31] et a mis l'accent sur la proposition de nouvelles normes pour les communications dans les réseaux à faible consommation d'énergie auxquels appartiennent la plupart des communications M2M, en adaptant le protocole IPv6 sur la pile de protocole 802.15.4.
Un autre groupe de travail de l'IETF, Routing Over Low power and Lossy networks (ROLL), a développé un protocole de routage pour les réseaux à faible puissance et à perte (LLN : Low-power and Lossy Network), qui sont constitués de nombreux dispositifs intégrés dotés de ressources limitées et interconnectés par divers liens, tels que l'IEEE 802.15.4, le Bluetooth et le WiFi à faible consommation d'énergie.
ZigBee [32] est une norme de réseau sans fil avec un rayon de couverture relativement faible, mais dont la fiabilité est assez élevée avec une consommation réduite. La spécification ZigBee définit la couche réseau et la couche application sur la base de la norme 802.15.4 PHY/MAC (couches Physique et MAC).

4.3. OneM2M

OneM2M est un consortium dont la mission est d'harmoniser les normes IoT/M2M en développant une spécification M2M mondialement reconnue, qui réponde au besoin d'une couche de service M2M commune avec une vue indépendante de l'accès aux services de bout en bout. [33]
Cette spécification porte sur les exigences, l'architecture fonctionnelle, les solutions de sécurité et les interfaces avec les protocoles IoT les plus couramment utilisés CoAP, MQTT et HTTP.
L'architecture à trois couches de oneM2M comprend des applications, une couche de services communs (middleware) et des réseaux.

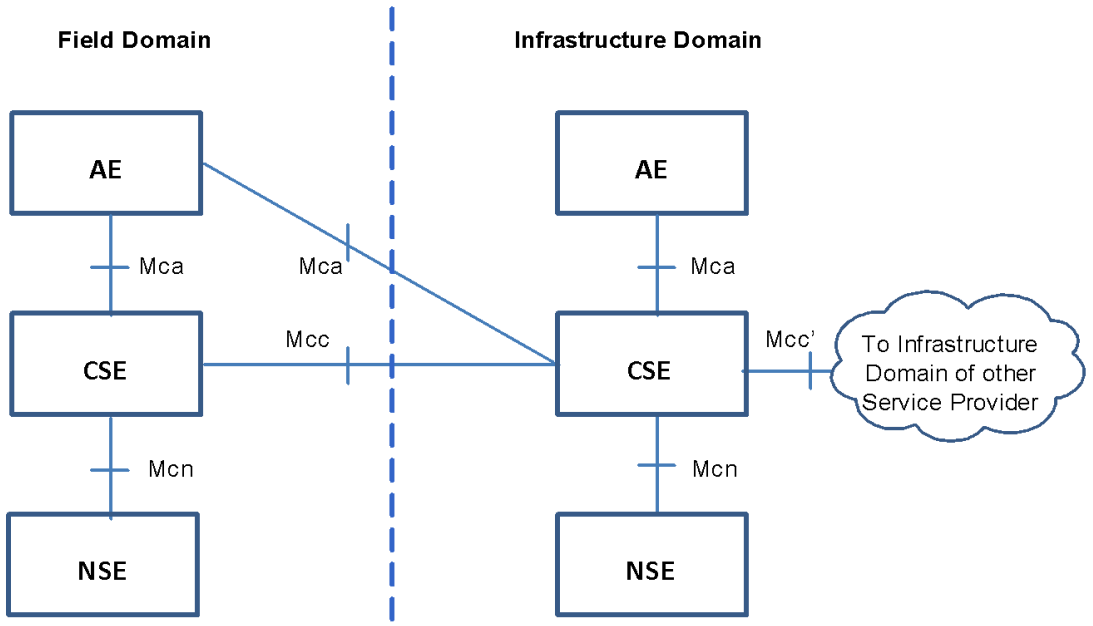


Figure 2.3. Architecture OneM2M : Vue fonctionnelle

L'architecture fonctionnelle de oneM2M présentée dans la Figure 2.3. comprend les entités suivantes [34] :

- Entité d'application (AE) : L'entité d'application est une entité de la couche d'application qui met en œuvre la logique d'un service d'application M2M. Parmi les exemples d'AE, on peut citer une instance d'application de suivi de flotte, une application de mesure de la glycémie à distance, une application de mesure de l'énergie ou une application de contrôle d'un processus industriel.
- Entité de services communs (CSE) : Une entité de services communs représente une instanciation d'un ensemble de fonctions de services communs de la couche de services du oneM2M. Parmi les exemples de fonctions de service offertes par la CSE, on peut citer : le stockage et le partage de données avec contrôle et autorisation d'accès, la détection et la notification d'événements, la communication de groupe, la programmation d'échanges de données, la gestion d'appareils et les services de localisation.

- Entité de services de réseau sous-jacente (NSE) : Une entité de services de réseau fournit aux ECS des services provenant du réseau sous-jacent. Parmi ces services, on peut citer les services de localisation, le déclenchement d'appareils, certains modes de veille comme le PSM dans les réseaux basés sur le 3GPP.

4.4. LWM2M : Protocole de communication M2M Standardisé

Le besoin de normalisation au niveau services M2M et applications a emmené les organisations et en particulier l'Open Mobile Alliance (OMA) à la définition du LWM2M (lightweight M2M) [35]. Ce standard définit le protocole de communication de la couche application entre un serveur LwM2M et un client LwM2M (objet ou device M2M) en permettant à la fois la standardisation du format de sortie des données et des actions de gestion de périphériques.

Ce protocole est léger vu la faible quantité de données transférées entre les appareils et les plates-formes de gestion. Il peut fonctionner à n'importe quelle vitesse de connexion. Il est basé sur le protocole de transfert COAP et peut supporter dans de nouvelles versions de messagerie les protocoles MQTT et sur HTTP. Il utilise les protocoles DTLS et TLS pour assurer la sécurité des communications au niveau transport.

Le schéma de la Figure 2.4. décrit l'architecture LWM2M composée de 4 interfaces logiques (API) utilisées pour établir une communication entre un serveur LWM2M et un client LWM2M (un capteur par exemple) :

- Démarrage (Bootstrapping) : Cette interface permet au serveur de bootstrap LWM2M de fournir au client LWM2M les informations lui permettant de s'enregistrer auprès d'un serveur LWM2M : clé de sécurité, contrôle d'accès et configuration du produit.
- Enregistrement (Registration) : Cette interface permet à un Client LWM2M de s'enregistrer auprès du serveur LWM2M et de signaler les fonctionnalités du produit au serveur LWM2M.
- Accès à un Objet ou une ressource (Object Resource Access) : Cette interface permet au serveur LWM2M d'accéder à la ressource d'une instanciation d'un objet (OIR : Objet Instance Ressource) du Client LWM2M.
- Rapport (Reporting) : Cette interface permet au Client LWM2M de signaler au serveur LWM2M, les modifications périodiques ou sur évènement des OIR.

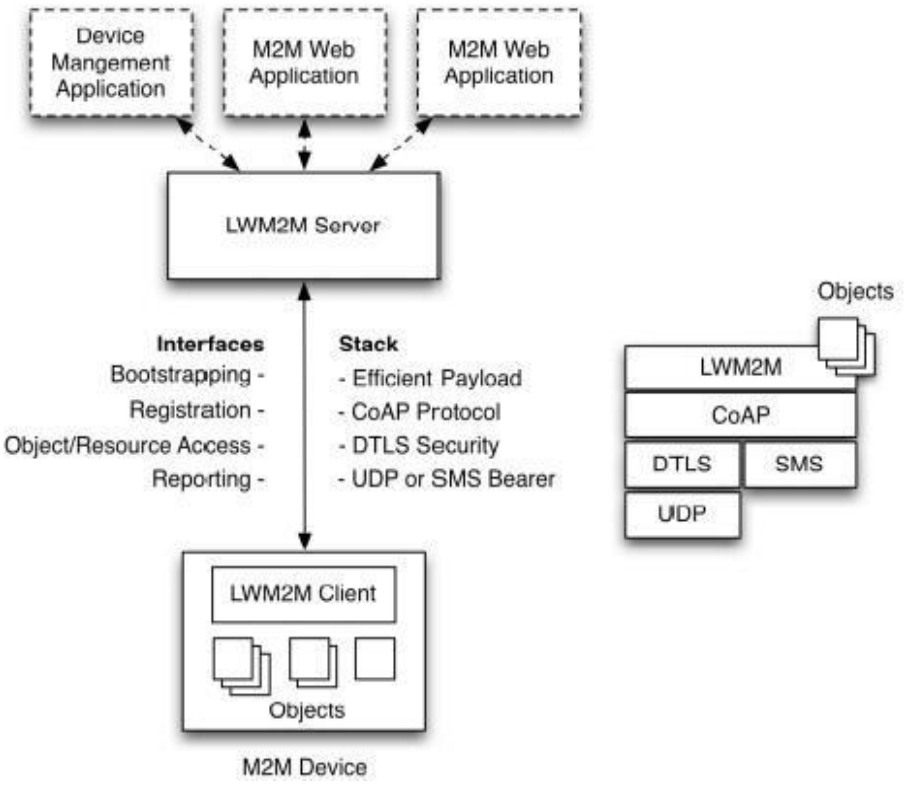


Figure 2.4. Architecture LWM2M

5. Principales caractéristiques des communications M2M

Selon la norme 3GPP, les communications M2M se distinguent par plusieurs caractéristiques, notamment une faible consommation d'énergie, un faible débit, une tolérance temporelle, la transmission de petites quantités de données, une faible mobilité etc. [36] [37]. Les principales caractéristiques sont énumérées et résumées ci-dessous.

- Faible mobilité : les dispositifs M2M ne se déplacent pas, se déplacent rarement ou se déplacent uniquement dans une certaine région. Cette fonction permet à l'opérateur de réseau de simplifier et de réduire la fréquence des procédures de gestion de la mobilité.
- Contrôle du temps : envoi ou réception de données uniquement à certaines périodes prédéfinies ce qui peut donc éviter toute signalisation inutile en dehors de ces instants.
- Tolérance temporelle : Le transfert de données peut être retardé, ce qui permet à l'opérateur de réseau d'empêcher les dispositifs M2M qui sont tolérants au temps d'accéder au réseau (par exemple, en cas de surcharge du réseau d'accès radio).
- Commutation par paquets : l'opérateur de réseau peut fournir un service de commutation par paquets avec ou sans MSISDN.
- Origine mobile uniquement (Mobile Originated Only) : Cette fonction est destinée aux applications dans lesquelles il est possible de réduire la fréquence de la gestion de la mobilité. Le réseau devrait être en mesure de fournir un mécanisme permettant à l'opérateur du réseau, de configurer dynamiquement les dispositifs M2M, pour qu'ils n'exécutent les procédures de gestion de la mobilité qu'au moment des communications d'origine mobile.

- Petites transmissions de données en ligne : Les dispositifs MTC envoient ou reçoivent fréquemment de petites quantités de données.
- Surveillance : Cette fonction permet de surveiller l'état des dispositifs M2M. Le but n'est pas d'empêcher le vol ou le vandalisme, mais de fournir une fonctionnalité permettant de détecter les événements.
- Message d'alarme prioritaire : La fonction M2M de message d'alarme prioritaire peut être utilisée avec des dispositifs M2M qui émettent une alarme prioritaire en cas de mauvais fonctionnement, vol, sabotage d'autres besoins nécessitant une attention immédiate
- Connexion sécurisée : Cette fonction M2M est appliquée aux dispositifs M2M qui nécessitent une connexion sécurisée pour les communications entre les dispositifs M2M et le(s) serveur(s) M2M.
- Déclencheur spécifique à l'emplacement : Cette fonction est destinée aux applications dans lesquelles les dispositifs M2M sont signalés positionnés dans une zone précise en utilisant les informations de localisation.
- Transmission peu fréquente : Cette fonction est destinée aux dispositifs M2M dont l'intervalle entre deux transmissions de données est long. Le réseau ne doit fournir une ressource qu'au moment de la transmission effective.
- Faible consommation énergétique : les dispositifs MTC sans fil sont alimentés par batterie.

6. Applications M2M

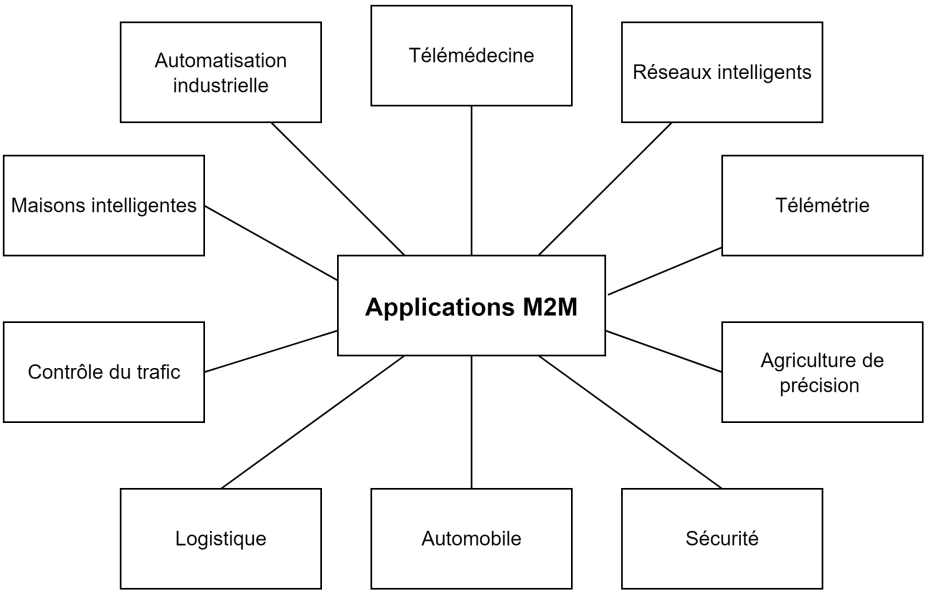
Ces dernières années, le M2M a trouvé un grand nombre d'applications dans de nombreux secteurs illustrés dans la Figure 2.5, notamment la télématique, l'automatisation industrielle, la télésurveillance, le transport intelligent, les soins de santé, la sécurité, la gestion du parc automobile, les points de vente, les compteurs intelligents, les maisons intelligentes, les réseaux intelligents, etc.

Réellement, il existe de nombreuses applications M2M, nous donnons ici la description de quelques applications concrètes [5] [38].

- L'utilisation la plus classique du M2M est la télémétrie. Les communications M2M peuvent être utilisées pour facturer les utilisateurs en fonction de leurs consommations (gaz, eau, électricité) à travers des compteurs intelligents. Les machines M2M peuvent être utilisées aussi dans les usines pour détecter des conditions telles que la pression, la température et l'état des équipements.
- Technologie des réseaux intelligents : Cette application M2M permet aux services publics de surveiller et de gérer plus efficacement les réseaux publics (par exemple, le réseau de transport), en collectant des données à partir de capteurs intelligents et d'autres dispositifs.
- Gestion de la flotte automobile : La technologie M2M est utilisée pour suivre et gérer les flottes de véhicules, aidant les entreprises à optimiser les itinéraires, à améliorer le rendement énergétique et à réduire les coûts de maintenance.
- Télémédecine : La technologie M2M est utilisée pour surveiller les patients à distance, ce qui permet aux prestataires de soins de santé de suivre les signes vitaux et

d'autres paramètres de santé sans que les patients soient obligés de se rendre dans un hôpital ou une clinique.

- Automatisation industrielle : La technologie M2M est utilisée dans les environnements industriels pour automatiser les processus et améliorer l'efficacité. Par exemple, des capteurs peuvent être utilisés pour surveiller les équipements et détecter les problèmes potentiels avant les pannes.
- L'industrie automobile utilise la technologie M2M basée sur l'IA, ce qui permet aux consommateurs de conduire leur voiture en mode mains libres. Le mécanisme d'utilisation d'un véhicule à conduite autonome est le suivant est qu'il est équipé de capteurs qui surveillent en permanence tout ce qui se passe autour du véhicule, avec l'utilisation d'applications d'IA pour prendre les bonnes décisions. Les capteurs installés dans le véhicule lui permettent de capturer des milliers de données à chaque milliseconde et l'utilisation de l'IA permet aux données d'agir en conséquence de manière rapide. Les capacités IoT d'une automobile permettent de se garer, de freiner et de changer de voie automatiquement, ce qui a un impact considérable sur l'avenir de la conduite.
- Maisons intelligentes : La technologie M2M est utilisée dans les systèmes domestiques intelligents pour automatiser l'éclairage, le contrôle de la température, la sécurité... etc. Les systèmes domestiques intelligents intègrent la communication M2M pour permettre également aux appareils et aux équipements domestiques, de communiquer entre eux et d'envoyer des messages par l'intermédiaire d'un réseau.
- Gestion des distributeurs automatiques : Par exemple, un distributeur automatique peut avertir le commerçant en cas de rupture de stock d'un produit.
- Agriculture de précision : Dans le secteur agricole, divers capteurs sont utilisés pour surveiller à distance la température, l'humidité, l'ensoleillement et les précipitations pluviales. Cela permet d'augmenter la productivité et de réduire les coûts de production.



- **Figure 2.5.** Applications M2M.

7. Plateformes M2M

Une plateforme M2M est une solution logicielle qui unifie et simplifie la gestion des appareils et des applications M2M. Une plateforme M2M peut avoir un large éventail de caractéristiques et de fonctionnalités, mais dans l'ensemble, elle gère les données transmises par les appareils, les systèmes dorsaux qui traitent les données, la fourniture de mises à jour logicielles aux appareils et l'administration générale du cycle de vie des appareils [2].
Il existe trois types de plateformes de services M2M [38] :

- Les plateformes des dispositifs connectés : sont des éléments logiciels qui facilitent le déploiement et la gestion des appareils connectés pour les applications M2M sur les réseaux cellulaires.
- Les plateformes de mise en œuvre d'applications : Elles sont conçues pour gérer les appareils ou les machines en fournissant les fonctions de base pour plusieurs applications M2M. Elles permettent les activités d'extraction et de normalisation des données, de sorte que les applications M2M et les systèmes d'entreprise peuvent facilement exploiter les données des machines.
- Les plateformes de développement d'applications : Elles fournissent une couche de service et des API normalisées pour les développeurs d'application

Le marché des plateformes M2M devrait se développer avec une croissance exponentielle au cours de la période de prévision de 2023-2030. La croissance du marché peut être attribuée à la demande croissante de la plate-forme M2M en raison du transport, de l'énergie, de la santé, de la vente au détail, des applications de sécurité au niveau mondial [39].
Il n'existe pas de plateforme normalisée pour les appareils dans la technologie M2M. Plusieurs plateformes M2M commercialisées ont été conçues pour être spécifiques à une tâche ou à un appareil.

Parmi celles-ci on peut citer : ThingWorx [40], Everyware cloud [41],.AT&T M2M Application Platform [42],.Etherios Cloud Connector [43].SensorCloud [44].Everythng [45], etc.
Les caractéristiques de ces plateformes logicielles peuvent être résumées comme suit [46] :

- Protocole d'application et interface : La plupart des plateformes utilisent le protocole d'accès aux objets simples (SOAP) pour les interfaces et l'architecture RESTful basée sur le protocole HTTP. Certaines utilisent le protocole CoAP allégé, et d'autres le protocole HTTPS pour des raisons de sécurité.
- Enregistrement : Avant toute utilisation, les appareils et les utilisateurs doivent être enregistrés sur la plateforme.
- Source ouverte : Les développeurs d'applications et les utilisateurs peuvent développer de nouveaux services en utilisant le kit du développeur de logiciel (SDK) fourni par la plateforme.
- Traitement des données : La plateforme peut traiter et analyser les informations collectées à partir des appareils, telles que les valeurs maximales, minimales et moyennes des données, ou effectuer d'autres tâches, les valeurs maximales, minimales et moyennes des données ou exécuter d'autres fonctions personnalisées.

- Compte : Les plateformes fournissent des comptes différenciés en fonction de la confidentialité, du stockage, du type de service (SMS, service web, nuage) et du nombre d'appareils disponibles.
- Accès de l'utilisateur : Les plateformes prennent en charge les appareils mobiles (applications), portails web ou programmes d'application
- Prise en charge du réseau M2M : Certaines plateformes permettent aux passerelles de connecter des appareils qui ne peuvent pas se connecter directement à Internet.
- Plateforme pour entreprise : Certaines plateformes sont destinées aux entreprises ou à des fins publiques ; elles permettent la prise en charge d'un grand nombre de nœuds, par exemple pour l'automatisation industrielle ou la surveillance de l'environnement.

Il est à remarquer que contrairement à l'utilisation de plateformes prêtes à l'emploi d'autres plateformes de développement peuvent être conçues et réalisées par les utilisateurs eux-mêmes, à l'exemple d'Arduino.

8. Exigences des communications M2M

Comme le décrit l'ETSI, l'institut européen de normalisation des télécommunications, les exigences qui doivent être satisfaites dans la conception du système de gestion du réseau M2M sont résumées comme suit dans [1] [18].

- Scalabilité (extensibilité) : le M2M doit communiquer sans erreur malgré la connexion d'un plus grand nombre d'objets.
- Anonymat : le M2M ne doit pas divulguer le nom de l'utilisateur et les informations le concernant lorsqu'une demande est faite, à moins qu'il ne soit soumis à des exigences spécifiques.
- Journalisation : Le M2M doit enregistrer les activités importantes, par exemple le nombre de tentatives infructueuses lors de l'installation, les informations sur les défauts et les services non fonctionnels. Les journaux sont mis à la disposition de l'utilisateur sur demande.
- Tolérance aux pannes : Gérer les nœuds morts (épuisement de l'énergie, dommages physiques).
- Coût et complexité : Faible coût, faible complexité
- Dépendance à l'égard de l'application : Application centrée sur les données, application d'urgence ou application en temps réel
- Efficacité énergétique : Batterie limitée ; recharge difficile ou impossible
- Configuration dynamique : Nœuds morts, nouveaux nœuds et nœuds mobiles
- Minimiser le trafic de gestion : Le trafic de gestion peut affecter les performances du réseau

9. Défis posés par les communications M2M.

La présence d'un nombre énorme de dispositifs M2M pose des problèmes de communication critiques, notamment en ce qui concerne l'extensibilité (scalabilité), l'hétérogénéité, les problèmes d'accès aux ressources radio, la sécurité des communications, l'efficacité

énergétique, la prise en charge de la qualité de service (QoS), la résistance aux pannes ou aux dysfonctionnements des appareils.

9.1. Scalabilité des réseaux M2M

Le nombre de dispositifs qui croit engendre des problèmes d'accès tels que les interférences, saturation des ressources radios… les applications M2M sont des applications à large bande exploitant des réseaux cellulaires, car elles transmettent des séquences de paquets de petite taille (température, humidité…) alors que les réseaux cellulaires ont été conçus pour répondre aux besoins des applications humaines (H2H), qui sont généralement des applications à large bande [47].
Pour répondre au problème de scalabilité, une première solution a consisté à augmenter la capacité du réseau en construisant davantage de pico cellules avec des stations de base et des cellules de taille réduite ou en passant à des technologies LTE de plus grande couverture. Une deuxième solution consiste en l'utilisation des technologies de communication réseau complémentaires (telles que WiFi, WiFi-Direct, Zigbee, Bluetooth, etc.), pour acheminer le trafic de données mobiles, initialement prévu pour être transmis sur des réseaux cellulaires [48].
En outre, il faut considérer les défis inhérents aux limites des infrastructures matérielles et logicielles mises en place pour le support des communications sans fil (H2H et M2M) qui se partagent les mêmes ressources de réseaux de communication. Différentes solutions ont été adoptées pour pallier à cette contrainte [49] ; entre autres, l'utilisation d'une technologie réseau de proximité, en l'occurrence l'Edge Computing pour éviter la surcharge du réseau de communication, et l'utilisation des technologies de la 5G, telles que la communication D2D (Device to Device), qui exploitent les échanges P2P (Peer to Peer) et participent ainsi à la solution de ce problème.

9.2. Communication M2M Pair à Pair

Initialement, dans les applications M2M, les données sont envoyées à une plateforme de services M2M. Les utilisateurs et les appareils communiquent par l'intermédiaire de la plateforme pour accéder aux données stockées ; ils ne peuvent pas communiquer directement sans passer par la plateforme. Ce type de communication n'est pas efficace en termes de délai.
La communication Pair à Pair présente des avantages. En plus d'alléger la surcharge des stations de base et des réseaux de communication, et participer à régler le problème de scalabilité comme il a été souligné dans le paragraphe précédent, la communication P2P peut réduire les délais et empêcher le passage inutile et le stockage du trafic par la plateforme [50].
La communication P2P permet de traiter les données M2M localement, lorsque les dispositifs M2M qui s'échangent des messages, sont proches les uns des autres. Les dispositifs terminaux peuvent aussi communiquer directement par l'internet sans passer par la plateforme, cela inclut la communication directe (D2D) d'appareil à appareil entre les nœuds terminaux. Cet avantage s'accentue avec l'augmentation du nombre de dispositifs M2M et d'applications nécessitant une communication P2P [2] [51].

9.3. L’efficacité énergétique

Le clustering est une technique avantageuse à plus d’un titre. Elle permet de limiter l’accès à la station de base et éviter les problèmes d'encombrement et de surcharge, en désignant quelques nœuds, appelés chefs de groupe (Cluster Heads ou CHs), comme relais pour les autres nœuds terminaux. De cette manière, le nombre de demandes d'accès à la station de base est limité au nombre de CHs. En outre, une sélection appropriée des CHs peut contribuer à réduire la consommation d'énergie du système en exploitant les transmissions multi-sauts sur des liaisons à gain élevé au lieu de transmissions directes sur de plus grandes distances [43]. La solution à ce niveau réside dans la conception d’approches efficaces en énergie pour l'élection des CHs et la formation de clusters avec l’affectation des différents dispositifs M2M aux différents clusters.

9.4. La congestion dans les communications M2M

Du fait de l’accélération technologique dans l’IoT, une augmentation exponentielle du trafic M2M est plus que probable [17]. Par conséquent, une capacité de données importante devient un défi sérieux pour tous les opérateurs de réseaux afin d'éviter le phénomène de congestion à de différents niveaux de l’architecture M2M.

- ·Niveau d'accès : Un nombre excessif d'appareils implique l'envoi d'un très grand nombre de demandes d'accès, ce qui entraîne de nombreux problèmes de surcharge qui se traduisent par des échecs de tentatives d'accès.
- Niveau du réseau : Le nombre considérable de dispositifs M2M censés envoyer leurs données utiles simultanément nécessite un épuisement rapide de la bande passante, ce qui est l'une des principales préoccupations de l'opérateur pour dimensionner tout réseau qui vise à faire évoluer les dispositifs M2M dans son plan futur.
- Niveau de l'application : Le troisième niveau de gestion du trafic pourrait être un moyen de prévenir la congestion du réseau, en donnant aux opérateurs de réseau la possibilité de donner la priorité au trafic de données des différentes applications M2M, et à imposer des politiques d'utilisation du réseau garantissant la satisfaction de l’utilisateur.

9.5. Qualité de Service (QoS)

Les réseaux de communication M2M sont chargés de transmettre, selon l’application, des données en temps réel ou en différé en respectant les conditions relatives aux différents paramètres de QoS à savoir, le débit, la bande passante, la fiabilité, la disponibilité, la latence, la gigue, la sécurité et le respect de la vie privée. Cependant, les applications M2M ont des exigences différentes en matière de qualité de service [37].

Par exemple, Il existe des applications sensibles aux retards telles que les applications de télémédecine, alors que pour les applications multimédias, c’est la bande passante, ainsi que la gigue et le délai, qui sont les paramètres les plus importants à assurer, d’autres applications sont tolérantes au retard, mais nécessitent une grande fiabilité [52].

Si les applications M2M partagent les mêmes challenges et solutions que les applications Internet au niveau des réseaux de communication, les protocoles de routage au niveau des

réseaux des dispositifs M2M, qui ont leurs propres contraintes, présentent plusieurs défis de QoS, outre le délai et le taux de livraison des paquets, l'overhead de routage (c'est-à-dire le nombre de paquets de routage transmis par paquet de données et le temps supplémentaire nécessaire pour leurs transmission) doit être également pris en compte.

9.6. Hétérogénéité

Les dispositifs M2M peuvent provenir de divers constructeurs et utiliser différentes technologies. Cette hétérogénéité peut rendre difficile la mise en œuvre de mesures de sécurité cohérentes pour tous les appareils [53]. L'une des principales conditions du succès des communications M2M dans les réseaux 5G est leur capacité à intégrer de nombreux types d'appareils de différentes capacités énergétique, de calcul et de stockage, d'énergie, et de communication (débit, latence, etc.).
En outre, le système M2M doit pouvoir prendre en charge des applications dont les caractéristiques et les exigences peuvent être extrêmement différentes. Ces propriétés d'hétérogénéité du système global font de la conception des protocoles de communication une tâche très difficile [37]. Les caractéristiques du trafic M2M sont différentes de celles du trafic existant d'homme à homme et d'homme à machine.
En raison de l'hétérogénéité des objets connectés en termes de capacité et de trafic généré, la gestion du trafic M2M doit être adaptée en fonction des exigences de l'application et du modèle de trafic [54].

9.7. Autres défis

Enfin, d'autres défis et difficultés qui restent sujets ouverts à la recherche dans le domaine des communications M2M : [37][55]

- Gestion du spectre : Dans les communications M2M sans fil, la rareté du spectre est un problème sérieux. Par conséquent, l'utilisation efficace du spectre dans un environnement de partage devrait être soigneusement étudiée.
- Accès d'opportunité : La technique de la radio cognitive est utilisée pour détecter les trous dans le spectre et les utiliser pour un accès dynamique. Cette technique est souple et permet de prendre en charge différents systèmes, y compris les applications M2M dans le cadre de la LTE/LTE-A. Cependant, l'accès d'opportunité nécessite des technologies complexes pour détecter l'espace blanc du spectre et des protocoles efficaces de gestion des ressources radio sans interférer avec les utilisateurs principaux.
- Connectivité : pour les réseaux 5G et au-delà, la taille du réseau ne cesse d'augmenter pour répondre à la demande d'un grand nombre de dispositifs M2M. Il convient de mentionner qu'un système trop connecté devient difficile à gérer (par exemple, en raison de l'interférence excessive), il sera donc important d'arrêter ce qui doit être connecté afin de fournir les capacités de communication nécessaires pour les dispositifs M2M et pour éviter les interférences entre le grand nombre de cellules et les stations de base eNB et gNB.

En outre, les normes LTE/LTE-A, 5G du 3GPP fournissent un accès sans fil omniprésent aux stations de base (eNB : Evolved Node Base Stations in 4G LTE Networks or gNB : next generation Node Base Station) par des liaisons à saut unique dans les communications H2H.

Toutefois, l'hétérogénéité et le grand nombre de dispositifs dans les communications M2M peuvent créer des conflits et des problèmes aux communications H2H. [56]
Par conséquent, les réseaux 5G, pour éviter l'utilisation d'un seul saut dans les communications M2M, prévoient la possibilité de communications D2D multi-sauts.

9.8. Sécurité des communications M2M :

Enfin, nous terminons ce chapitre par l'une des principales préoccupations concernant les communications M2M. La sécurité dans le contexte des communications M2M représente un défi important pour les chercheurs, en raison des complexités introduites par le large déploiement, la décentralisation et l'hétérogénéité des appareils dans les réseaux des dispositifs M2M. Les solutions de sécurité conventionnelles ne sont pas adaptées aux systèmes M2M, principalement en raison des contraintes imposées par les appareils à ressources limitées. En outre, les dispositifs M2M fonctionnent sans l'intervention humaine. Tout cela conduit directement à une augmentation des risques de sécurité tels que le vol, le sabotage, l'écoute, les intrusions, violations de données, les accès non autorisés, etc.
Cette préoccupation est la problématique de cette thèse. Elle fait l'objet des chapitres suivants qui porteront sur les solutions publiées dans la littérature, et les solutions que nous suggérons comme contributions dans le domaine.

10.Conclusion

Le M2M, notamment dans sa version sans fil, représente un sujet de grand intérêt pour l'avenir des TICs. Il s'agit d'une révolution technologique qui a un impact profond sur la vie quotidienne des humains en permettant la communication directe entre les machines.
Dans ce chapitre, nous avons présenté l'architecture générale du système de communications M2M en expliquant ses trois domaines. Certaines des caractéristiques M2M importantes ont été mises en évidence, ainsi que les défis attendus dans le futur. La scalabilité, l'hétérogénéité, la congestion, la qualité de service, la communication P2P, et d'autres challenges dont la sécurité, ont été abordés ainsi que les ébauches de solutions proposées dans la littérature pour les relever.
La sécurité représente un des défis majeurs dans les communications M2M. Le chapitre suivant portera en détail sur ce sujet en discutant les vulnérabilités, les attaques et les solutions pour sécuriser les communications M2M.

Chapitre 3

SÉCURITÉ DES COMMUNICATIONS M2M

1. Introduction

De nombreuses applications M2M dans différents domaines (militaire, santé, maison intelligente, industriel, etc.) traitent des informations confidentielles et vitales, nécessitant la mise en place de solutions de sécurité à différents niveaux, et essentiellement au niveau du domaine des réseaux des dispositifs M2M, du fait que les décisions prises sont basées sur les données générées par ces dispositifs.

La sécurité des données dans les réseaux de dispositifs M2M est d'autant plus préoccupante pour les raisons suivantes :

- Les communications seront de plus en plus sans fil, fonctionnant de manière autonome (sans intervention humaine)
- Les dispositifs ou appareils de prélèvement et de détection sans fil seront déployés à une grande échelle dans des lieux non protégés (cités, hangars, entreprises, forêts, montagnes, véhicules, chaînes d'approvisionnement...), ce qui rend les communications M2M vulnérables.

Le succès de ces applications dépend essentiellement des solutions de sécurité sur lesquelles elles reposent. Cependant, les protocoles de sécurité classiques sont généralement inappropriés pour la protection des dispositifs M2M, en raison de leurs ressources limitées en termes d'énergie, de stockage et de traitement.

2. Services de sécurité pour les communications M2M

Pour garantir la sécurité des données et des communications M2M, il faut tenir compte des contraintes imposées par les dispositifs M2M dans la mise en place des services de sécurité. Nous décrivons ici brièvement les services de sécurité qui doivent être assurés par les systèmes M2M, et qui consistent à protéger les systèmes, les réseaux et les programmes contre les cyberattaques, à savoir : [5][39][57]

- Confidentialité : Ce service garantit que seules les entités autorisées, disposant d'une clé de déchiffrement, peuvent lire les données relevées ou captées par un dispositif, au cours de leurs transmission chiffrée dans un système de communication M2M.
- Intégrité : Elle doit être assurée pour éviter toute altération des données (modifications non autorisées ou accidentelle, une suppression, ou une répétition de mots) prélevées ou collectées lors de leurs transmissions dans un système de communication M2M.
- Authentification : Elle permet de s'assurer que l'accès aux ressources des domaines des réseaux M2M, n'est permis que pour les dispositifs et utilisateurs autorisés.
- Non-répudiation : Elle garantit que les nœuds M2M ne peuvent pas nier la transmission après avoir envoyé des données.

- Disponibilité : Elle assure la pérennité des services et la disponibilité des ressources du domaine des réseaux M2M pour les applications M2M.
- Vie Privée (Privacy) : Elle permet de protéger la vie privée dans les systèmes de communication M2M critiques (exemple e-santé), et éviter que des informations sensibles ou privées ne soient divulguées illégalement.
- Fraîcheur : permet de s'assurer que les informations échangées dans un système de communication M2M sont récentes (fraîches) et qu'elles ne sont pas rejouées. En d'autres termes, un nœud M2M ne peut pas lire les messages précédemment transmis lorsqu'il rejoint le réseau.

3. Vulnérabilités des Communications M2M

Il existe des limitations à différents niveaux de l'architecture M2M, qui rendent les communications M2M vulnérables. Nous soulignons ici quelques vulnérabilités et défis de sécurité courants dans les communications M2M.
Comme présenté dans le chapitre précédent, l'architecture M2M est composée de trois domaines principaux (le domaine des dispositifs M2M, le domaine réseau de communication et le domaine application). Chaque domaine présente ses propres défis en matière de sécurité.

3.1. Au niveau du domaine réseau des dispositifs M2M

Le large déploiement des dispositifs M2M sans fil dans des endroits éloignés et sans surveillance, pose plusieurs défis de sécurité en plus des problèmes d'interférence et de la possibilité d'interception des communications sans fil. On peut citer les problèmes d'extension ou de scalabilité ainsi que le problème d'identification ou adressage, introduits dans le chapitre précédent. En plus, il est à noter le handicap posé par la vulnérabilité de ces dispositifs à la manipulation physique par des attaquants ; ces derniers peuvent voler ou endommager les dispositifs en modifiant leur logiciel ou même les composants matériels [5][58]. Il est donc impératif dans ce cas de mettre en œuvre des mesures de sécurité physique, y compris des installations sécurisées et du matériel inviolable.
Il est à noter aussi que la diversité des dispositifs M2M, qui utilisent différentes technologies et peuvent provenir de différents constructeurs, rend délicate la mise en œuvre de mesures de sécurité compatibles pour tous les appareils [55] [56].
En outre, les dispositifs M2M présentent aussi les contraintes de ressources, telles que la puissance de traitement et la durée de vie de la batterie. Il peut donc être difficile dans ce cas de mettre en œuvre des mesures de sécurité performantes.

3.2. Au niveau du domaine réseau de communication

Les réseaux de communication utilisés sont partagés avec d'autres applications H2H. Les réseaux ouverts et l'accès universel aux communications Internet sont fréquemment utilisés par les réseaux de communication M2M [59]. Les failles de sécurité posées par les protocoles ouverts tels que TCP/IP restent exploitables par les attaquants sur le réseau Internet, ce qui rend difficile d'avoir une visibilité particulière sur le trafic M2M, et donc complique la détection et la réponse aux attaques. [49]

En plus, les réseaux de communication M2M peuvent utiliser des technologies et des protocoles différents. Le problème d'interopérabilité se pose alors, il peut être difficile de mettre en œuvre des mesures de sécurité interopérables entre les différents réseaux.

3.3. Au niveau du domaine application

Les applications M2M collectent et traitent souvent des données sensibles. Les attaquants peuvent tenter de modifier, ou de supprimer ces données, afin de perturber le fonctionnement de l'application, ou d'avoir accès à des informations sensibles. Il faut s'assurer que seuls les utilisateurs autorisés et les appareils autorisés ont accès aux applications et aux données M2M. Il est difficile à ce niveau de détecter les comportements suspects. [49].
En plus, certaines applications M2M, comme la e-santé, ont des exigences strictes en matière de temps de latence. Les attaques par déni de service (DoS) peuvent avoir des répercussions désastreuses. Le déploiement de solutions de sécurité elles-mêmes peut entraîner des retards, ce qui nécessite des mesures de sécurité précises pour réduire les effets négatifs sur les applications. [60]

4. Attaques contre les Communications M2M

En plus des menaces classiques que les réseaux M2M partagent avec d'autres technologies ou d'autres réseaux, il existe d'autres menaces, particulières aux réseaux M2M, qui sont une combinaison de risques de sécurité provenant des caractéristiques inhérentes aux appareils et aux applications M2M, et des risques provenant de leur intégration avec les communications internet (H2H). L'hétérogénéité des réseaux qui sont combinés pour créer un réseau M2M et le nombre considérable de dispositifs interconnectés amplifient les dangers et la nécessité de proposer de nouveaux mécanismes sécurisés pour les communications.
Les attaques M2M peuvent se produire à différents niveaux (physique, réseau de communication et niveau application).

4.1. Attaques physiques

Les attaques physiques dans l'environnement du réseau M2M sont conçues pour détruire le matériel ou le logiciel d'un appareil. Elles peuvent être classées comme suit :

- Attaques par canal latéral. Une attaque par canal latéral exploite les propriétés physiques d'un appareil, telles que la consommation d'énergie ou le rayonnement électromagnétique, pour extraire des données sensibles. Par exemple, un pirate peut utiliser une attaque par canal latéral pour obtenir une clé de cryptage d'un appareil client. Dans un autre exemple, un pirate peut activer un processus de fabrication pour modifier un produit. [61] [62]
- Falsification de nœuds : Dans une attaque par altération de nœud, le pirate accède physiquement à un appareil et en prend le contrôle [63]. En manipulant physiquement l'appareil, l'attaquant peut accéder aux données contenues dans sa mémoire et effacer les données souhaitées.
- Modification du logiciel : Ce type d'attaque consiste à modifier le logiciel de l'appareil cible pour l'empêcher de fonctionner comme il le devrait [64]. Le contrôle sans fil est

une option pour cela. En produisant des données incorrectes ou invalides, le nœud touché met en péril l'intégrité des données. Les nœuds peuvent être sabotés à l'aide de cette technique d'attaque. Les attaquants peuvent utiliser cette méthode pour manipuler les paiements, comme le montrent les péages électroniques et les compteurs intelligents.

- Cheval de Troie matériel : un cheval de Troie matériel est une modification malveillante d'un lien matériel au cours de la fabrication. Ces modifications peuvent contenir du matériel malveillant ou contrôler les systèmes d'exploitation. Les chevaux de Troie matériels se produisent généralement lorsqu'un appareil est reçu d'une source non fiable [65].
- Dommages causés à l'équipement M2M : Étant donné que les équipements M2M sont souvent utilisés dans des zones facilement accessibles, ils sont vulnérables au vol ou aux dommages physiques.

4.2. Attaques logiques

Les attaques logiques dans les réseaux M2M visent à perturber la fonctionnalité du réseau sans nécessiter d'accès physique aux dispositifs cibles. Ces attaques exploitent les vulnérabilités logicielles et protocolaires pour compromettre la sécurité, l'intégrité et la disponibilité des communications M2M. Les attaquants peuvent être internes, ayant un accès légitime au réseau, ou externes, cherchant à infiltrer le système depuis l'extérieur. La diversité et l'interconnexion des dispositifs dans les réseaux M2M rendent ces attaques particulièrement préoccupantes, car elles peuvent causer des dommages significatifs à grande échelle. Les types d'attaques logiques incluent l'usurpation d'identité, les attaques par relais, les attaques contre les protocoles de routage et les violations de données.

4.2.1. Attaques Externes

Les attaques externes sont menées par des entités n'ayant aucun accès légitime au réseau M2M. Ces attaquants exploitent souvent les failles de sécurité à travers divers niveaux du réseau pour compromettre les dispositifs et les communications. Voici quelques exemples d'attaques externes :

1. **Espionnage (Eavesdropping)** : Une attaque passive et externe, où l'attaquant utilise d'abord l'espionnage pour découvrir les clés de communication utilisées par deux pairs, puis usurpe l'identité d'une des parties en manipulant les messages et leur flux, contrôlant ainsi totalement les communications. L'attaque de l'homme du milieu (MitM) est une attaque réseau courante influencée par les communications basées sur Internet.
2. **Capture de Nœud (Node Capture)** : Une attaque active et externe où l'attaquant, au lieu d'intercepter les communications ou de chercher une faille dans la sécurité du réseau, prend (manuellement) le contrôle d'un appareil pour en extraire des informations, plutôt que de le détruire.
3. **Déni de Service Distribué (DDoS)** : Dans une attaque DDoS, un flux continu de demandes provenant de multiples nœuds malveillants endommage le fonctionnement

du réseau et perturbe les communications. Ces requêtes de flooding peuvent cibler différentes couches de la pile protocolaire, créant différents types d'attaques qui nécessitent une attention particulière. Par exemple, l'attaque DDoS qui perturbe les communications sans fil en brouillant le signal par l'envoi de données et de requêtes inutiles est considérée comme une attaque au niveau physique. Au niveau de la couche MAC (Medium Access Control), une attaque DDoS implique que les nœuds malveillants envoient des paquets en même temps qu'un nœud légitime, entraînant des collisions de paquets et une diminution des performances du réseau [66].

4. **Fuzzing** : Dans cette attaque, l'objectif de l'attaquant est de provoquer la défaillance d'un appareil ou d'une application en utilisant des messages générés ou manipulés aléatoirement. Les attaquants peuvent insérer des exploits spécifiques dans les messages, tels que des débordements de tampon, des caractères de format spécial ou des données d'entrée invalides, pour trouver des erreurs d'implémentation dans une application ou un service de l'appareil.

4.2.2. Attaques Internes

Les attaques internes sont effectuées par des entités faisant partie du réseau. Elles peuvent être particulièrement difficiles à contrer car l'attaquant a déjà un certain niveau de confiance et d'accès au système. Voici quelques exemples d'attaques internes :

1. **Usurpation d'Identité** : Dans les attaques par usurpation d'identité, l'attaquant se fait passer pour un utilisateur autorisé du réseau. Ces attaques incluent également les attaques par rediffusion, où l'attaquant capture et retransmet des données pour se faire passer pour un autre utilisateur.
2. **Attaque par Relais** : Un attaquant interne peut intercepter les communications entre deux appareils, une source et une destination, puis, sans déchiffrer le message intercepté, il l'utilise ultérieurement et au moment opportun pour communiquer avec le deuxième appareil. L'exemple le plus connu de cette attaque est l'interception du signal émanant d'une télécommande pour ouvrir une porte de voiture ou d'un local.
3. **Attaques contre les Protocoles de Routage** : Ces attaques visent à manipuler les décisions de routage le long des routes de communication. Par exemple, les attaques byzantines, les attaques par trou de ver et les attaques Sybil permettent à un attaquant de prendre plusieurs identités et de causer des dommages importants [67].

4.3. Violation des données

Les attaques sur les données sont généralement effectuées par le biais d'une technologie d'écoute (sniffing). Dans ce cas, les attaques peuvent être actives ou passives selon que l'action consiste à altérer ou tout simplement prendre connaissance de données sensibles sur la vie privée ou le comportement de l'utilisateur, par exemple, des informations sur sa santé, ses habitudes …etc. Des exemples de ces attaques sont les suivantes :

- Études du trafic: Les attaquants utilisent le 'reniflement' de trafic pour examiner les transmissions et identifier les communicants. L'attaquant surveille passivement le trafic entre deux ou plusieurs nœuds communicants pour collecter et découvrir des

informations pertinentes. Cette attaque est relativement facile, car la communication sans fil est une transmission radio ouverte et ne perturbe pas la communication M2M.

- Attaque de l'homme du milieu (MitM) : Dans cette forme d'attaque active, l'auteur se place entre deux dispositifs ou entre un serveur et un dispositif, pour intercepter les messages et les échanger entre les deux communicants, après les avoir modifiés selon le besoin.
- Attaque d'intégrité : Lorsque des données sont transmises, conservées dans la mémoire d'un appareil ou hébergées sur un serveur d'application, les attaques d'intégrité peuvent compromettre leur intégrité. L'attaquant introduit de fausses informations lors d'une attaque d'intégrité. La manipulation des données détectées ou des informations de localisation erronées peut parfois avoir des effets néfastes dans certaines applications.
- Transfert sélectif : Les attaques par transfert sélectif consiste en la sélection de certains paquets pour les relayer, les autres paquets sont supprimés du réseau. Les attaques par trou noir (Blackhole) est un exemple d'attaque par transfert sélectif. Dans une attaque par trou noir, les nœuds malveillants rejettent tous les paquets au lieu de les transmettre. Dans une attaque par trou gris (Grayhole), les nœuds malveillants abandonnent certains paquets tout en en transmettant d'autres [49]. Les attaques par transfert sélectif peuvent être utilisées pour perturber la communication M2M et compromettre la confidentialité, l'intégrité et la disponibilité des données. Par exemple, un attaquant peut utiliser une attaque par trou noir pour empêcher la transmission des données des capteurs à un serveur central, ou une attaque par trou gris pour corrompre les paquets de données en transit.

Pour maintenir l'intégrité et la sécurité des réseaux M2M, il est essentiel de lutter contre ces menaces. La mise en œuvre de mécanismes d'authentification robustes, de protocoles de communication sécurisés, de contrôle d'accès et de technologies de cryptage permet de réduire les risques liés à ces attaques.

Nous résumons dans la Table 2.1, les attaques sur les communications M2M selon le type et le service de sécurité concerné.

Table 2.1: Attaques sur les réseaux M2M.

Attaques	Type d'attaque	Vulnérabilité	Exigence de sécurité
Attaques par canal latéral	Physique, Externe, Passive	Sécurité physique	Confidentialité
Falsification de nœud (Node Tampering)	Physique, Passive Interne/Externe	Communication sans fil, sécurité physique, contrainte de ressources	Confidentialité, Authentification
Modification du logiciel	Physique, Externe, Active/Passive	Sécurité physique, contrainte de ressources, logiciel	Authentification, intégrité, disponibilité
Trojans matériels	Physique, Active/Passive, Interne	Sécurité physique	Confidentialité, intégrité
Destruction du dispositif M2M	Physique, Active, Externe	Sécurité physique	Intégrité, disponibilité
Spoofing	Logique, Active, Interne/Externe	Communication sans fil, Contrainte de ressources, logiciel, Connexion globale, protocole standard ouvert	Authentification
Déni de service (DoS)	Logique, Active, Interne/Externe	Communication sans fil, contrainte de ressources, logiciel	Disponibilité
Attaques par relais	Logique-Active/Passive, Interne/Externe	Communication sans fil, logiciel, contrainte de ressources, sensibilité au délai	Authentification, intégrité
Attaques par protocole de routage (ex : sybil attack)	Logique, Active, Interne	Contrainte de ressources, sensibilité au retard, scalabilité	Disponibilité, intégrité
Transmission selective (ex: black hole attack)	Attaque de données, Active, Interne	Connexion globale, sensibilité aux délais, scalabilité	Authentification, intégrité, disponibilité
Écoute clandestine (Eavesdropping)	Attaque de données, Passive, Externe	Communication sans fil, contrainte de ressources, globalité de la connexion	Confidentialité
Man in the middle	Attaque de données, Active/Passive, Interne/Externe	Communication sans fil, contrainte de ressources, globalité de la connexion	Confidentialité, Authentification
Analyse du trafic	Attaque de données, Passive, Externe	Communication sans fil	Confidentialité
Attaque d'intégrité	Attaque de données, Interne	Contrainte de ressources, communication sans fil	Intégrité
Transmission sélective	Active-Attaque de données, interne	Connexion globale, sensibilité aux délais, scalabilité	Authentification, intégrité, disponibilité

5. Solutions de sécurité pour les communications M2M

Dans la littérature, on retrouve plusieurs recherches [1][3][4][5], qui ont étudié diverses solutions pour contrer les vulnérabilités des communications M2M, et répondre à leurs exigences en matière de services de sécurité (confidentialité, contrôle d'accès, etc…gestion des clés, protection de la vie privée, intégrité, authentification) au niveau application et au niveau réseau des dispositifs M2M.

5.1. Sécurité M2M au niveau Application

La sécurité des applications garantit que seules les instances approuvées d'une application peuvent communiquer entre elles, tandis que les instances illégitimes ne peuvent pas interférer.

Au niveau du domaine application, les protocoles de communication M2M/IoT les plus utilisés sont : Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Simple Sensor Interface (SSI) and Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protocol (XMPP) [68] [69].

En général à ce niveau, les solutions de sécurité sont basées sur les protocoles de sécurité classiques de l'internet tels que : SSL/TLS, DTLS, IPSEC (IP Security) et HIP (Host Identity Protocol).

Des solutions de sécurité basées sur le protocole MQTT, le plus utilisé actuellement pour les communications IoT/M2M, et qui a été conçu pour être léger, flexible et simple à mettre en œuvre, donc adéquat pour les objets connectés qui sont limités en ressources. L'authentification qui consiste à la vérification de l'identité d'un utilisateur ou d'un appareil spécifique connecté au broker MQTT. Chaque connexion doit passer par au moins un pare-feu qui met en œuvre des règles sophistiquées pour l'accès. Pour assurer la sécurité des communications à ce niveau, le protocole fait appel aux protocoles TLS (Transport Layer Security) et SSL (Secure Sockets Layer), qui sont à la base des protocoles cryptographiques qui établissent un canal de communication sécurisée entre un client et un serveur, après négociation de divers paramètres de connexion. Une fois la poignée de main terminée, une communication cryptée assurant la confidentialité entre le client et le serveur est établie.

Des versions sécurisées du protocole SMQTT (Secure MQTT) et MQTT-SN (MQTT for Sensor Networks) ont été proposées, basées sur le chiffrement léger à l'aide de la cryptographie légère à courbe elliptique [68] [70].

AMQP : Advanced Message Queuing Protocol (protocole avancé de mise en file d'attente des messages) . Il s'agit d'un protocole centré sur les messages, qui s'ajoute à TCP/IP et qui permet de publier, de s'abonner et de communiquer de pair à pair. AMQP prend en charge la communication orientée message par le biais de garanties de livraison des messages. En termes de sécurité, il prend en charge l'authentification SASL et TLS pour la communication sécurisée des données. [71].

CoAP : Constrained Application Protocol est un protocole de transfert web qui prend en charge les requêtes unicast et multicast pour une utilisation dans des dispositifs et réseaux contraints. Il est basé sur une architecture demande-réponse entre les nœuds communicants. Les messages sont échangés par UDP. CoAP fournit une sécurité via le Datagram Transport Layer Security (DTLS) qui est un protocole sécurisé pour le trafic réseau permettant de gérer

la perte de paquets. Un schéma léger de CoAP sécurisé pour l'Internet des objets baptisé Lithe a été proposé. Lithe montre que DTLS peut être allégé, et que sa surcharge peut être considérablement réduite [72].
Il existe d'autres protocoles de communication IoT/M2M tel que XMPP qui utilise le protocole TLS avec l'extension StartTLS pour le cryptage des canaux, ce qui protège le flux contre la falsification et l'écoute clandestine. Avant d'utiliser SASL pour préserver la confidentialité des informations d'identification, il est nécessaire de procéder à un shakehand complet de la session TLS [73].

5.2. Solutions de sécurité du réseau des dispositifs M2M sans fil

L'un des principaux défis des communications M2M est lié à la sécurité des dispositifs M2M, qui devraient fonctionner sans intervention humaine et sont donc sans défense. Ces dispositifs sont donc exposés à la manipulation physique par des attaquants. Les attaquants peuvent voler ou endommager les dispositifs, ou tenter de modifier le logiciel ou le matériel des dispositifs. Des mesures de sécurité physique sont donc à mettre en œuvre en prévoyant des installations sécurisées et de faire en sorte que le matériel soit inviolable.
En outre, l'hétérogénéité des technologies et des systèmes mis en œuvre sur ces dispositifs les rend plus vulnérables et rend difficile leur protection contre diverses attaques.
La plupart des recherches dans ce domaine sont basées sur des solutions de sécurité traditionnelles pour assurer les différents services de sécurité, et qui restent inappropriées pour les dispositifs M2M.
Au bas niveau (couche du réseau de perception ou des dispositifs) dans les architectures M2M et IoT, différents protocoles de sécurité ont été proposés sur la base de schémas cryptographiques symétriques classiques, ou d'ECDH (Elliptic Curve Diffie-Hellman Key Exchange) asymétriques légers [74], ou encore basée sur des opérations XOR, ou des fonctions de hachage légères [75].
Nous présentons dans ce qui suit, quelques travaux et approches suggérées dans la littérature pour assurer les services de sécurité dans les réseaux des dispositifs M2M sans fil.

5.2.1. Confidentialité

Le réseau des dispositifs M2M sans fil est un domaine où les ressources sont limitées, les appareils utilisés sont de petite taille avec une capacité de calcul limitée, une mémoire limitée et des batteries de très faible puissance. Les algorithmes cryptographiques classiques ne peuvent pas être appliqués à des fins de sécurité.
Ces appareils intelligents à ressources limitées ont donné naissance à un nouveau domaine appelé cryptographie légère (LWC). Les algorithmes de cryptage légers sont conçus pour être efficaces et légers, tout en offrant une sécurité élevée. Ils conviennent donc aux dispositifs M2M à ressources limitées.
L'AES-GCM et le ChaCha20 sont des exemples d'algorithmes de chiffrement légers [61]. Ces algorithmes peuvent être utilisés pour protéger diverses communications M2M, telles que la transmission de données entre les dispositifs M2M et les serveurs Cloud, la communication entre les dispositifs M2M et d'autres dispositifs sur le réseau, et les mises à jour logicielles pour les dispositifs M2M. Un système M2M doit disposer de fonctions telles que la gestion à

distance ou les mises à jour du micrologiciel pour se protéger des cyberattaques et des tentatives illégales de connexion.
Au niveau du réseau des dispositifs M2M, de nombreux algorithmes de chiffrement par blocs légers (PRESENT, LEA, HIGHT, TEA, CLEFIA [76][77] [78], et des fonctions de hachage légères (PHOTON, QUARK et SPONGENT) [79], ont été proposés dans la littérature pour les dispositifs limités en ressources (RFID tag, capteurs sans fil, …etc.).

5.2.2. Gestion des clés de chiffrement

La gestion des clés est essentielle pour la sécurité et la fiabilité des communications M2M. Les clés sont la base de l'authentification, de la confidentialité et de l'intégrité dans les systèmes
M2M. Les stratégies actuelles de gestion des clés M2M utilisent souvent le chiffrement à clé publique (PKC) pour générer des clés en toute sécurité entre les dispositifs. Les solutions sont les suivantes

- L'infrastructure à clé publique (PKI) : Pour que les dispositifs M2M authentifiés puissent recevoir et transmettre des données cryptées, cette solution dépend d'une autorité de certification (CA), un tiers fiable. [80]
- Gestion symétrique des clés : Cette solution utilise une clé partagée entre les deux systèmes de communication pour sécuriser la transmission des données. Les clés sont généralement générées à l'aide de systèmes d'échange de clés sécurisés tels que la cryptographie Diffie-Hellman ou les courbes elliptiques. [74]

5.2.3. Protection de la Vie Privée (Privacy)

La protection de la vie privée est importante pour les solutions de sécurité M2M, car elle protège les informations sensibles contre toute divulgation non autorisée. Les solutions adoptées reposent en général sur les techniques de contrôle d'accès et de chiffrement, en plus du recours à l'instauration de la technique d'anonymat, qui remplace ou supprime les identifiants personnels des données, afin d'empêcher l'identification d'une entité spécifique
[81].
Une approche générale du problème de la protection de la vie privée consiste à introduire de l'entropie dans le système. Le concept d'entropie consiste à quantifier le caractère aléatoire du système, de sorte que l'on peut se rendre compte de la difficulté pour l'attaquant de voler les informations personnelles. Comme pour la randomisation de l'adresse MAC, la puissance d'émission peut également être « randomisée » afin d'empêcher l'attaquant d'estimer la distance de l'appareil le plus proche en utilisant la fonction RSSI.[82]

5.2.4. Contrôle d'accès

Les systèmes de contrôle d'accès limitent l'accès aux données aux seuls utilisateurs autorisés. Pour ce faire, ils mettent en œuvre des mécanismes d'authentification et d'autorisation des utilisateurs, qui garantissent que seules les personnes ou les dispositifs autorisés peuvent accéder aux informations sensibles [5]. La sécurité M2M repose aussi sur diverses techniques de contrôle d'accès, notamment le contrôle d'accès basé sur les rôles (RBAC), les réseaux

privés virtuels (VPN) et le pare-feu. Ces contrôles servent à restreindre l'accès aux communications M2M, en veillant à ce que les données sensibles ne soient accessibles qu'aux parties disposant d'une autorisation appropriée

5.2.5. Intégrité

Les solutions d'intégrité sont essentielles à la sécurité M2M pour garantir l'exactitude et la fiabilité de la transmission des données entre les appareils. La modification illicite des données peut avoir de graves conséquences, en particulier dans les domaines d'application M2M critiques (e-santé, contrôle de processus industriel, etc.) [4].
Le maintien de l'intégrité des données est vérifié pour s'assurer que les messages transmis n'ont pas été altérés. L'opération repose sur l'utilisation de signatures numériques. Avant d'envoyer une communication à un autre dispositif, une signature numérique ou un MAC (code d'authentification de message) est ajouté au message pour garantir son intégrité. Généralement, la signature est produite par l'utilisation de fonctions de hachage pour générer une empreinte ou un condensat ou encore un MAC qui sera transformé en signature numérique après son chiffrement par une clé privée de la source. Ce qui permet à la destination ayant accès à la clé publique de la source de vérifier l'authenticité de ce dernier, en déchiffrant la signature et en la comparant au condensat ou au MAC du message reçu par le destinataire.

5.2.6. Authentification

Les recherches dans le domaine de la sécurité M2M se sont intéressées particulièrement aux services d'authentification et de l'intégrité des données. L'authentification est une condition préalable à la sécurité des communications M2M. Elle nécessite des approches et des infrastructures appropriées permettant aux stations de base, ou à des serveurs de confirmer l'identité des nœuds et l'authenticité des données captées ou échangées entre les dispositifs M2M. Les contraintes imposées par le manque de ressources des dispositifs M2M, nécessitent d'implémenter des solutions qui minimisent le volume et le nombre de messages échangés.
Dans un réseau M2M, l'authentification peut se faire à l'aide d'un secret ou d'une clé pré-partagée (PSK) implémentée sur le dispositif ou d'un certificat numérique délivré par une autorité, l'autorité de certification (CA). Un code d'authentification est utilisé pour vérifier qu'un message ou un paquet de données provient effectivement d'une certaine source. Pour ce faire, l'expéditeur signe un document à l'aide d'un numéro privé et le destinataire vérifie la signature à l'aide de la clé publique de l'expéditeur. L'information peut être vérifiée à l'aide d'autres techniques telles que les codes d'authentification des messages (MAC) et les algorithmes de hachage.
Plusieurs protocoles d'authentification pour les communications M2M ont été publiés, nous présentons dans la suite quelques protocoles, que nous qualifions de traditionnels, et qui utilisent des techniques classiques, et d'autres plus récents basées sur de nouvelles approches.

5.2.6.1. Protocoles d'authentification traditionnels

SPINS [83] est considéré comme le premier protocole de sécurité qui répond à diverses exigences de sécurité des réseaux de communication sans fil, principalement au niveau de l'appareil. Pour crypter les messages et générer des codes d'authentification des messages, SPINS utilise une version allégée de RC5. Il assure des services de sécurité séparément sur deux protocoles pour les communications unicast et multicast.

TinySec [84] et Minisec [85] sont des protocoles d'authentification fonctionnant respectivement au niveau de la couche liaison et de la couche réseau. Ils sont considérés comme des protocoles d'authentification efficaces, particulièrement bien adaptés aux nœuds limités. Tinysec est le premier protocole de sécurité entièrement mis en œuvre pour les WSN, basé sur le code d'authentification des messages (MAC) et la chaîne de blocs de chiffrement (CBC). Son inconvénient est lié à l'utilisation d'une clé unique pour générer les codes MAC.

En 2012, Chen et al [86] ont proposé un protocole d'authentification de groupe et d'accord de clé appelé G-AKA. Dans leur schéma, un réseau de capteurs (SN) peut authentifier un groupe de stations mobiles (MS) avec l'aide du réseau domestique (HN). Chen et al., affirment que le système qu'ils proposent est sûr, et qu'il peut satisfaire à toutes les exigences de sécurité. Toutefois, leur protocole est vulnérable aux attaques de type man-in-the-middle, DoS et de redirection.

En 2013, Lai et al [87] ont proposé un protocole d'authentification de groupe et d'accord de clé, appelé SE-AKA. Plus précisément, SE-AKA utilise la technique de Diffie-Hellman à courbe elliptique (ECDH) pour réaliser et garantit la confidentialité et le secret des clés (KFS/KBS), et adopte également un système de cryptage à clé asymétrique pour protéger la vie privée des utilisateurs. Pour l'authentification de groupe, il simplifie l'ensemble de la procédure d'authentification en calculant une clé temporaire de groupe (GTK).

Plus tard dans [88], Giustolisi et al., ont proposé un protocole d'authentification de groupe et d'accord de clé (GROUP-AKA) conçu pour la communication M2M. Le protocole tend à réduire la latence et la consommation de bande passante, et à tenir compte de l'extensibilité ou la mise à l'échelle à un grand nombre d'appareils. Le protocole est considéré léger car il repose uniquement sur le cryptage à clé symétrique, il est donc compatible avec les appareils à faible ressources.

Cependant, le protocole utilise des clés asymétriques pour l'authentification, ce qui entraîne un surcoût de calcul élevé.

Dans [89], les auteurs ont présenté un autre protocole d'authentification de groupe et d'accord de clé pour les dispositifs M2M dans les réseaux 3GPP, appelé GLARM. Ce protocole d'authentification mutuelle et d'accord de clé sécurisée a été proposé pour les dispositifs à ressources limitées et comporte deux phases principales : la phase d'initialisation et la phase d'authentification de groupe et d'accord de clé. Le protocole consiste en deux parties permettant d'obtenir une authentification de groupe efficace et sécurisée dans le cas d'un accès 3GPP et d'un accès non 3GPP, respectivement.

Une étude plus récente [90] propose un protocole léger d'authentification et de distribution de clés (LAKD) pour les communications M2M. Il permet à un appareil de s'assurer de l'identité d'un autre appareil et distribue des clés à utiliser pour la confidentialité et l'intégrité des

données échangées. LAKD est conçu pour les appareils à ressources très limitées et repose sur les opérations légères XOR, addition et soustraction, et une fonction de hachage.
Dans [91], les auteurs ont proposé dans cet article un mécanisme d'authentification léger, basé uniquement sur des opérations de hachage et l'opération logique XOR, pour les communications M2M de machine à machine entre un dispositif industriel à ressources limitées (par exemple, un capteur intelligent) comprenant un élément sécurisé (SE) et un routeur équipé d'un module de plateforme de confiance (TPM : Trusted Platform Module).
Le mécanisme d'authentification comprend deux procédures :
1) La procédure d'enregistrement où le capteur est enregistré auprès du serveur d'authentification (AS)
2) La procédure d'authentification où l'authentification mutuelle entre le capteur et le routeur est réalisée.
Le mécanisme proposé est caractérisé par un faible coût de calcul, une faible surcharge de communication et de stockage, tout en réalisant l'authentification mutuelle.
De nombreux protocoles ont été proposés pour sécuriser les communications entre les capteurs, les actionneurs et d'autres dispositifs pour les réseaux de dispositifs M2M. La gestion de l'identité et de l'authentification des dispositifs interconnectés peut être assurée par une autorité centrale basée sur l'infrastructure à clé publique (ICP). Malheureusement, les méthodes basées sur IPSec et SSL /TLS ne sont pas adaptées aux appareils à ressources limitées. D'autres recherches [92],[93] se sont concentrées sur le 6LoWPAN (IPv6 Low Power Wireless Personal Area Network) développé par l'IETF Internet Engineering Task Force dans la RFC 6568.
Dans [93], les auteurs proposent une authentification mutuelle et un établissement de clés pour les communications M2M dans les réseaux 6LoWPAN. Le protocole applique l'algorithme ECDH (Elliptic Curve Diffie-Hellman) pour mettre en œuvre la distribution des clés secrètes entre les nœuds.
Sur la base de l'analyse du protocole EAKES6Lo, qui montre son défaut dû à la perte de données après la capture du nœud et sa vulnérabilité aux attaques par trou d'air et aux attaques par choix de texte en clair, un nouveau protocole d'authentification mutuelle pour la communication M2M basé sur 6LoWPAN est proposé dans [86]. Ce protocole établit un mécanisme raisonnable de distribution des clés secrètes et conçoit une méthode de détection des attaques anti-capture pour les nœuds non surveillés afin de résister aux attaques telles que les attaques par rejeu, les attaques par trou d'air, les attaques par choix de texte en clair et les attaques par capture physique.
SAKES (Secure Authentication and Key Establishment Scheme) est proposé dans [94]. Il est basé sur une cryptographie à clé publique légère utilisant le schéma ECC afin d'établir la clé de session. Celle-ci est établie entre le 6LoWPAN et le serveur, sur la base d'un échange Diffie Hellman. Pour réduire la charge de calcul des nœuds de capteurs, SAKES effectue la plupart des calculs au niveau des nœuds passerelles et envoie la clé calculée aux nœuds de capteurs dans l'environnement 6LoWPAN.

5.2.6.2. Approches basées IA pour sécuriser les communications M2M dans l'IoT

L'intelligence artificielle (IA) et l'apprentissage machine (ML) facilitent la communication entre les systèmes, leur permettant de faire leurs propres choix de manière autonome ; ils

peuvent être utilisés pour répondre aux attaques de sécurité en temps réel en offrant une approche proactive pour la sécurité des communications M2M.
Les systèmes de détection d'intrusion IDS/IPS, utilisent de plus en plus l'apprentissage automatique et l'intelligence artificielle, pour améliorer leurs capacités de prévention en développant des signatures d'attaques sophistiquées afin de pouvoir détecter des anomalies qui seraient difficiles à découvrir avec des méthodes traditionnelles [5].
Les applications IoT/M2M génèrent un énorme volume d'informations. Avant que les données ne soient utilisées, elles doivent faire l'objet d'un processus de vérification automatique afin d'éviter toute donnée malveillante ou redondante. L'utilisation de techniques d'apprentissage ou ML (Machine Learning) peut permettre aux appareils IoT/ M2M d'évaluer les activités et les délais au sein des applications IoT/M2M afin de détecter les logiciels malveillants [95].
De manière générale, l'apprentissage automatique permet de résoudre certains problèmes de sécurité tels que la détection des intrusions, la préservation de la vie privée et le contrôle d'accès. En fait, plusieurs algorithmes d'apprentissage supervisé sont largement utilisés pour résoudre les problèmes de sécurité tels que le contrôle d'accès [96] [97], et la détection des intrusions [98] [99].
En outre, l'apprentissage supervisé peut être utilisé pour prendre des contremesures contre les attaques de sécurité dans l'IoT/M2M, dont les suivantes : Système de détection d'intrusion, Détection des logiciels malveillants, Détection des anomalies, Identification des appareils IoT non autorisé, Déni de service distribué ou DDoS, attaque par brouillage, attaque par usurpation d'identité.

5.2.6.3. Blockchain pour sécuriser les communications M2M dans l'IoT

La blockchain peut constituer un moyen sûr et infalsifiable de stocker et de transmettre des données entre machines, ce qui renforce encore la sécurité M2M.
La blockchain est une nouvelle méthodologie innovante de plus en plus utilisée par le monde universitaire et industriel en raison de ses diverses caractéristiques avantageuses en tant que base de données ou registre distribué, décentralisé et immuable maintenu dans un réseau peer-to-peer [6][7][8][9].
Malgré ses nombreux avantages, elle présente plusieurs limites car elle nécessite de grandes ressources de calcul, un espace de stockage important, une grande évolutivité, etc.., qui entravent sa mise en œuvre dans le monde réel, en particulier pour les dispositifs M2M dans notre cas. C'est pourquoi la blockchain est combinée à d'autres technologies telles que le Cloud, l'Edge ou le Fog Computing pour surmonter ses limites.
Dans le reste de ce manuscrit, nous proposons une solution basée sur la blockchain au niveau Edge proche du domaine des dispositifs M2M, pour sécuriser les communications dans les réseaux de dispositifs M2M tout en réduisant au maximum les limites de la blockchain.

6. Conclusion

La sécurité est d'une importance capitale pour les communications M2M en raison de l'importance de leurs applications dans des domaines critiques (contrôle de processus industriels, télémédecine, etc…). Les communications M2M font face à des défis posés par le déploiement étendu, la décentralisation et l'hétérogénéité des appareils M2M. Les solutions de sécurité classiques rencontrent des difficultés quant à leur implémentation sur des dispositifs contraints en ressources.

Les services de sécurité que doit assurer un système M2M, ainsi que les attaques potentielles auxquelles il doit faire face à différents niveaux de son architecture (serveurs d'application, réseaux de communication, et réseau des dispositifs M2M) ont été énumérées et discutées. L'examen des solutions de sécurité M2M, a été axé sur l'identification des risques et des vulnérabilités potentiels associés aux communications M2M.

L'utilisation de mécanismes de chiffrement légers, pour assurer les services de confidentialité et d'intégrité des données, ont été également abordées. Une attention particulière a été accordée au service d'authentification qui représente la pierre angulaire d'une sécurité efficace. Un état de l'art a alors été dressé, présentant des systèmes d'authentification pour les communications M2M/IoT proposés dans la littérature, distinguant les solutions d'authentification basées sur des techniques classiques, des solutions récentes qui utilisent des méthodes de l'intelligence artificielle.

Enfin, le chapitre a introduit la technologie blockchain que nous avons adoptée dans cette thèse pour sécuriser les communications M2M. Le chapitre suivant s'intéresse avec plus de détails au concept de blockchain et ses applications, en insistant sur son apport et sa prédisposition à assurer la sécurité des systèmes informatiques, et en particulier aux communications M2M.

Chapitre 4

BLOCKCHAIN POUR LA SÉCURITÉ DES COMMUNICATIONS M2M

1. Introduction

Dans ce chapitre, nous allons donner un aperçu général sur la technologie Blockchain et son application pour résoudre les problèmes de gestion et de sécurité dans le monde informatique. Historiquement, Introduite par Haber et Stornetta [100], la blockchain a suscité un intérêt intense grâce au principe du Bitcoin énoncé par Nakamoto en 2008 [101]. Le bitcoin, la crypto-monnaie numérique, a été le premier cas d'utilisation de la blockchain, permettant aux transactions commerciales et financières d'être effectuées de manière anonyme et sans l'intervention des banques. La monnaie numérique Bitcoin est utilisée par les utilisateurs impliqués ou les participants à une transaction comme forme acceptée (astuce ou moyen) d'échange. La force du bitcoin est qu'il ne nécessite pas de tiers de confiance. Par conséquent, la blockchain a suscité une grande attention pour être appliquée à de nombreux autres domaines et secteurs, et pas seulement à la monnaie bitcoin.

2. Concept de la Blockchain

Les blockchains sont une technologie numérique émergente qui combine la cryptographie, la gestion des données, les réseaux et plusieurs mécanismes de vérification, d'exécution et d'enregistrement des transactions entre différentes entités. Un registre blockchain ('ledger') est une liste (' chaîne ') de groupes ('blocs') de transactions.

Les entités qui proposent une transaction peuvent l'ajouter à un ensemble de transactions destinées à être enregistrées dans le ledger, les nœuds de traitement prennent certaines de ces transactions, vérifient leurs intégrités et les enregistrent dans de nouveaux blocs sur le ledger. Ce dernier sera répliqué à travers et dans de nombreux nœuds géographiquement répartis. Ces nœuds travaillent conjointement sans le contrôle d'une partie tierce de confiance. Néanmoins, La technologie garantit que tous les nœuds parviennent finalement à un consensus en termes d'intégrité et de contenu du ledger de la blockchain.

Traditionnellement, les transactions de valeurs entre différentes entités, sont effectuées à l'aide d'une partie tierce de confiance comme une banque ou autres. La blockchain fournit un concept différent pour réaliser ces transactions: au lieu de faire confiance à une seule partie tierce, la confiance est faite désormais grâce au travail collaboratif des nœuds de traitements dans le réseau blockchain. La Figure 4.1. illustre le fonctionnement de la technologie blockchain.

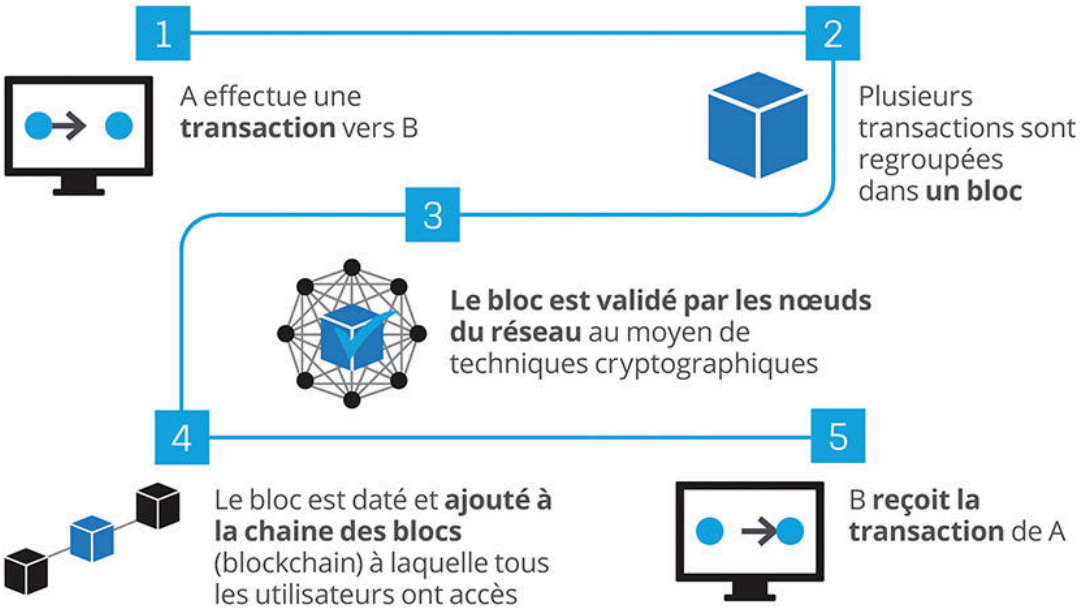


Figure 4.1. Fonctionnement principal de la blockchain.

Plusieurs définitions ont été attribuées à la technologie Blockchain, nous donnons ici une définition non-technique de ce paradigme :
La Blockchain est un grand livre digital qui est stocké sur plusieurs nœuds dans un réseau public ou privé (Figure 4.2). Les Blockchain comprennent des enregistrements de données ou des blocs. Chaque transaction est placée dans un bloc au fur et à mesure. Chaque bloc est relié à celui qui le précède. Ce dernier est ajouté de façon irréversible et les transactions sont “bloquées” ensemble - d'où le terme "blockchain". Une fois que ces blocs sont rassemblés dans une chaîne, ils ne peuvent être modifiés ou supprimés par un acteur. Au contraire, ils sont vérifiés et gérés à l'aide de protocoles de consensus [102].

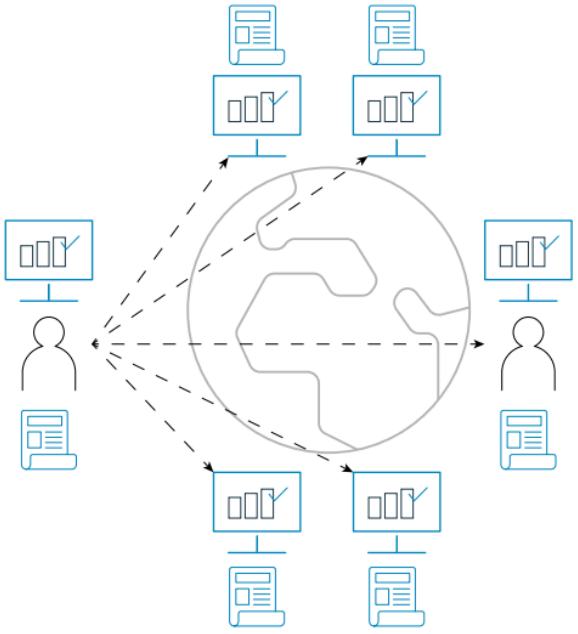


Figure 4.2. Le monde de la Blockchain : Un réseau distribué

3. Les protocoles de consensus

Un algorithme de consensus est un accord utilisé dans un réseau décentralisé pour prendre collectivement une décision lorsqu'elle est nécessaire. Ses propriétés comprennent la non-répudiation, l'authentification, le contrôle décentralisé, la transparence et la tolérance aux fautes byzantines[103]. Les auteurs dans [104] ont décrit les cinq composantes de l'algorithme de consensus : (1) proposition de bloc, (2) validation de bloc, (3) propagation de l'information, (4) finalisation de bloc et (5) mécanisme d'incitation.

Une fois que le nouveau bloc est généré à travers le mineur, ce dernier va propager ce bloc pour ses voisins dans le réseau blockchain. Cependant, il se peut que le mineur rencontre un bloc compétitif d'un autre mineur, ce cas est résolu grâce au mécanisme de consensus défini par la blockchain [105].

Effectivement, l'approche du consensus est fixée par la blockchain utilisée, nous pouvons par exemple voir que le protocole Bitcoin utilise le consensus Preuve de travail (PoW pour Proof of Work en Anglais). Ethereum [106] utilisait aussi le consensus Preuve de travail et a migré vers le consensus Preuve d'enjeu (PoS pour Proof of Stack en Anglais). HyperLedger Fabric quant à lui casse un peu les règles en proposant de choisir librement entre les différents consensus mais qui utilise principalement le consensus de Tolérance pratique aux fautes byzantines (PBFT pour Practical Byzantine Fault Tolerance en Anglais), Le consensus DPoS (Delegated Proof-of-Stake en Anglais) et Raft. Nous allons définir les consensus les plus populaires ainsi que voir leurs principales différences.

3.1. Preuve de travail

La preuve de travail (ou Proof of Work) est une stratégie de consensus utilisée dans le cadre du protocole Bitcoin [101]. Dans un réseau décentralisé un nœud doit être sélectionné pour enregistrer une transaction. Le moyen le plus simple est la sélection aléatoire. Cependant, la sélection aléatoire est vulnérable aux attaques. Ainsi, si un nœud veut publier un bloc de transactions, il faut beaucoup de traitement par ordinateur pour prouver que le nœud n'est pas susceptible d'attaquer le réseau.

Dans PoW, chaque nœud du réseau calcule une valeur de hachage de l'en-tête du bloc. L'en-tête de bloc contient un nonce et les mineurs changent fréquemment le nonce pour obtenir des valeurs de hachage différentes. Le consensus exige que le hachage soit égal ou inférieur à une certaine valeur donnée. Il exige que le résultat du calcul contient un certain nombre de zéros à gauche, spécifié par la valeur de difficulté (Target en Anglais). Cette valeur détermine la difficulté du calcul : plus le nombre de zéros requis est grand, plus il est difficile de trouver un résultat valide. Ce résultat est souvent représenté en hexadécimal, et la difficulté est ajustée pour que le résultat obtenu soit inférieur à la valeur cible, convertie en hexadécimal.

Lorsqu'un nœud atteint la valeur cible, il diffuse le bloc aux autres nœuds et tous les autres nœuds doivent confirmer mutuellement l'exactitude de la valeur de hachage. Si le bloc est validé, les autres mineurs ajoutent alors ce nouveau bloc à leur propre ledger. Les nœuds qui calculent les valeurs de hachage sont appelés "mineurs" et leur procédure dans un contexte PoW est appelée "minage" (mining en Anglais) dans le protocole Bitcoin.

Dans un réseau décentralisé, des blocs valables peuvent être générés simultanément lorsque plusieurs nœuds trouvent le nonce approprié presque en même temps. En conséquence, des branches peuvent être générées. Cependant, il est peu probable que deux fourches concurrentes génèrent le bloc suivant simultanément.
Dans le protocole PoW, la chaîne la plus longue par la suite est jugée comme étant la chaîne authentique.
Les mineurs doivent faire beaucoup de calculs en PoW, et ces derniers gaspillent trop de ressources. Pour limiter les pertes, certains protocoles de PoW dans lesquels les travaux pourraient avoir des applications secondaires ont été conçus. Par exemple, Primecoin [107] recherche des chaînes de nombres premiers spéciaux qui peuvent être utilisées pour la recherche mathématique [108]

3.2. Preuve d'Enjeu

La Preuve d'Enjeu, ou Proof of Stake (PoS), était introduite dans l'année 2011 sur le forum BitcoinTalk [109]. Le but de ce protocole de consensus est de réduire considérablement les dépenses énergétiques engendrées par la Preuve de Travail, tout en assurant une sécurité acceptable. Les mineurs seront remplacés par les validateurs, ces derniers sont sélectionnés en tenant compte du montant des jetons qu'ils possèdent ainsi que l'ancienneté des jetons qu'ils détiennent. Un validateur possédant le plus de jetons et que l'ancienneté de ces derniers est grande a donc beaucoup plus de chances d'être retenu pour valider un bloc[110].

3.3. Tolérance pratique aux fautes byzantines

Le consensus Tolérance pratique aux fautes byzantines (PBFT pour Practical Byzantine Fault Tolerance en Anglais) est un algorithme de réplication qui tolère les failles. Ce consensus peut gérer jusqu'à 1/3 des réplications byzantines malveillantes. Un nouveau bloc est déterminé lors d'une ronde. À chaque tour, une primaire serait sélectionnée selon certaines règles. Et il est responsable de l'ordre de la transaction. L'ensemble du processus pourrait être divisé en trois phases : pré-préparation, préparation et la validation (ou exécution). Dans chaque phase, un nœud entrerait dans la phase suivante s'il reçoit les votes de plus de 2/3 de tous les nœuds. Le PBFT exige donc que chaque nœud soit connu du réseau.

3.4. Le consensus de Preuve d'Enjeu Déléguée

Le consensus DPoS (Delegated Proof-of-Stake en Anglais) est considéré comme une variante de la preuve d'enjeu[110]. Il ne représente pas une amélioration significative, mais la différence entre la preuve d'enjeu (PoS) et le DPoS est principalement basée sur la démocratie directe, tandis que l'autre est basée sur une démocratie représentative. Dans le modèle DPoS, les mineurs (nœuds) se voient attribuer le droit de choisir leur représentant, appelé délégué. Le délégué est chargé d'accomplir trois tâches, notamment la création, la validation et la vérification du bloc. Le processus de validation serait beaucoup plus rapide si un nombre limité de mineurs (nœuds) effectuaient le processus de validation au lieu de l'ensemble du réseau. Par conséquent, cela aurait un impact direct sur le débit des transactions. De plus, les

délégués sont responsables de contrôler et de gérer la taille des blocs. Le délégué malhonnête ne devrait pas être une préoccupation car chaque nœud a le droit de voter pour le délégué de son choix. Bitshares est un exemple de mise en œuvre du DPoS.

3.5. Raft

Raft[111] est un algorithme de consensus conçu pour être facile à comprendre. Il est équivalent à Paxos en termes de tolérance aux pannes et de performance. La différence réside dans le fait qu'il est décomposé en sous-problèmes relativement indépendants, et il aborde de manière claire tous les principaux éléments nécessaires pour les systèmes pratiques. Le protocole Raft a été conçu par Diego Ongaro et John Ousterhout en 2013. Il est souvent comparé à l'algorithme de consensus Paxos en raison de sa similitude en termes de tolérance aux pannes et de performances. Cependant, Raft se distingue par sa structure modulaire et son approche plus intuitive, ce qui le rend plus facile à implémenter et à comprendre pour un large éventail de développeurs. Son objectif principal est de simplifier le processus de consensus, permettant ainsi à un plus grand nombre de personnes de créer des systèmes distribués robustes et fiables.

Table 4.1 : Comparaison des principaux mécanismes de consensus[112]

	PoW	**PoS**	**DPoS**	**Raft**
Scénarios d'application	Blockchain publique	Blockchain Privé	Blockchain Privé	Consortium
Degré de décentralisation	Entièrement	Entièrement	Entièrement	Semi-décentralisé
Nœud comptable	Réseau complet	Réseau complet	Nœuds sélectionnés	Basé sur le leader
Temps de réponse	Environ 10 minutes	Environ 1 minutes	Environ 3 secondes	Deuxième niveau
Débit	Environ 7 TPS	-	Environ 700 TPS	-
Efficacité de stockage	Ledger complet	Ledger complet	Ledger complet	Ledger complet
Tolérance aux pannes	50%	50%	50%	50%

4. Types de la Blockchain

La blockchain est généralement classée en trois catégories : la blockchain publique ou sans permission, la blockchain privée ou avec permission et la blockchain de consortium[113]. Nous comparons ces trois types de blockchain sous différents angles. La comparaison est présentée dans la Table 4.2.

4.1. Blockchain Publique

Dans une blockchain publique, il n'y a pas d'autorité dominante et aucune partie n'a plus de pouvoir que les autres dans le réseau. Les participants peuvent entrer et sortir à tout moment, et tout le monde peut participer au processus de consensus. Ces plateformes qui ne demandent aucune confiance sont sécurisées par les mécanismes de consensus que nous avons décrits ci-dessus.

4.2. Blockchain Privée

Avec une blockchain privée, une structure centralisée est suivie, où une seule entité a le plein pouvoir de valider les transactions et de prendre des décisions. La blockchain privée est plus efficace, facile à mettre en œuvre, utilise moins de ressources énergétiques et est plus rapide que la blockchain publique.
En réalité il existe deux catégories, publique et privée. Le consortium est un dérivé de la chaîne privée avec de multiples membres/participants identifiés et autorisés [112].

4.3. Consortium

Tous les membres n'ont pas les mêmes autorisations. Quelques membres du réseau de la blockchain se voient attribuer certains privilèges pour valider les nouveaux blocs. Les autres membres peuvent également valider les blocs, mais doivent parvenir à un consensus avant leur mise en œuvre.

Table 4.2 : Comparaisons entre la blockchain publique, consortium et privée[112]

Propriété	Blockchain publique	Blockchain consortium	Blockchain privée
Détermination du consensus	Tous les mineurs	Ensemble sélectionné de nœuds	Une organisation
Permission de lecture	Publique	Publique ou restreinte	Publique ou restreinte
Immutabilité	Presque impossible à altérer	Peut être altérée	Peut être altérée
Efficacité	Faible	Élevée	Élevée
Centralisé	Non	Partiel	Oui
Processus de consensus	Sans permission	Avec permission	Avec permission

5. Les plateformes Blockchain existantes

Dans cette partie nous présentons les plateformes les plus populaires de la blockchain, dont la liste se résume dans Bitcoin, Ethereum, Hyperledger Fabric, et quelques autres technologies que nous décrivons brièvement:
Bitcoin[101] est la première crypto-monnaie exploitée sur un réseau de type peer-to-peer. Contrairement aux systèmes bancaires et de paiement traditionnels, Bitcoin est basé sur une confiance décentralisée ; il n'y a pas d'autorité centrale de confiance dans le système. La confiance est due aux interactions entre les différents participants de l'écosystème. Dans le système Bitcoin, il existe un grand livre distribué qui stocke toutes les transactions Bitcoin depuis sa naissance en 2008. Le contenu du Ledger est répliqué dans de nombreux nœuds de traitement répartis géographiquement au sein du réseau Bitcoin.

Ethereum[106] est une plateforme pour la création et la publication des applications distribuées; plus fondamentalement, c'est une plateforme de crypto-monnaie générale de base qui est exécuté sur une machine virtuelle complète (ce qui signifie qu'elle peut exécuter n'importe quel script ou projet de crypto-monnaie). Plutôt que d'être une Blockchain, ou un protocole fonctionnant sur une Blockchain, Ethereum est une plateforme d'infrastructure sous-jacente fondamentale qui peut exécuter toutes les Blockchains et tous les protocoles, un peu comme une plateforme de développement universelle unifiée. Chaque nœud du réseau Ethereum fait fonctionner la machine virtuelle Ethereum pour une exécution transparente des programmes distribués (sous le nom de contrat intelligent ou Smart Contract en Anglais)[114].
Hyperledger est un effort de collaboration multi-projets à code source libre, hébergé par la Fondation Linux, créé pour faire progresser les technologies de Blockchain intersectorielles [115]. Hyperledger Fabric est un framework Blockchain destiné aux entreprises, pour le développement des solutions de base Blockchain avec une architecture modulaire. Les données peuvent être stockées dans plusieurs formats, et divers algorithmes de consensus peuvent être configurés, à savoir : le PBFT, le PoW et bien d'autres.
Tout comme l'Hyperledger Fabric, Corda[116] proposé par R3 a également des grands ledgers partagés uniquement entre des groupes d'entités bien définis. Ceci vise à améliorer la confidentialité et l'évolutivité en réduisant la réplication des données sur le réseau.
Ripple[117] est un système de règlement brut en temps réel, un réseau d'échange de devises et de transferts de fonds entre les institutions financières. Ripple utilise un ledger commun qui est géré par un réseau de serveurs de validation indépendants qui comparent constamment les enregistrements des transactions. Ces serveurs de validation peuvent appartenir à des particuliers ou à des banques.
Diverses techniques ont été proposées pour préserver la vie privée sur blockchain. Par exemple, Zcash[118] crypte les informations de paiement lors des transactions mais utilise une méthode cryptographique pour permettre à n'importe quel nœud de vérifier néanmoins la validité des transactions cryptées. Une construction à connaissance zéro est utilisée pour permettre au réseau de la Blockchain de maintenir le ledger sécurisé et de permettre un paiement privé sans divulguer les parties ou les montants impliqués.
IOTA est un nouveau système révolutionnaire à base Blockchain qui utilise une nouvelle innovation, appelée Tangle, en son cœur. Le Tangle est une nouvelle structure de données basée sur un graphique acyclique dirigé (DAG pour Directed Acyclic Graph en Anglais), ce dernier ni blocs, ni chaînes et ni de mineurs. En raison de cette nouvelle architecture radicale, le fonctionnement de l'IOTA est très différent de celui des autres Blockchains [119], ce réseau a pour but, à terme, de proposer des transactions sans frais, des transferts sécurisés ainsi qu'un nombre de transactions illimités.

Stellar[120] est une plateforme blockchain conçue pour faciliter les transferts de valeur de manière rapide, sécurisée et à faible coût. Fondée par Jed McCaleb en 2014, Stellar utilise le Stellar Consensus Protocol (SCP) pour permettre une validation efficace des transactions sans nécessiter de minage, ce qui réduit considérablement les frais et le temps de traitement. La cryptomonnaie native de Stellar, le Lumen (XLM), joue un rôle crucial en facilitant les échanges entre différentes devises et en évitant le spam du réseau. Contrairement à d'autres

blockchains, Stellar se distingue par son objectif de promouvoir l'inclusion financière mondiale en connectant les systèmes financiers, ce qui en fait une solution idéale pour les paiements transfrontaliers et les micro-transactions.

Tezos[121] est une plateforme blockchain décentralisée qui se distingue par son mécanisme de gouvernance en chaîne, permettant aux parties prenantes de voter sur les propositions d'amélioration du protocole. Fondée par Arthur et Kathleen Breitman, Tezos vise à créer une infrastructure dynamique capable d'évoluer et de s'améliorer au fil du temps sans nécessiter de hard forks. Elle utilise un mécanisme de consensus basé sur la preuve d'enjeu (Proof of Stake, PoS) et supporte des contrats intelligents écrits en Michelson, un langage formel conçu pour la vérification formelle des programmes. Cette vérification formelle améliore la sécurité des contrats intelligents, faisant de Tezos une option robuste pour les applications décentralisées.

Cardano[122] est une plateforme blockchain publique décentralisée qui repose sur une approche scientifique rigoureuse et sur une philosophie de recherche académique. Fondée par Charles Hoskinson, l'un des co-fondateurs d'Ethereum, Cardano vise à offrir une infrastructure de blockchain évolutive et durable pour les applications décentralisées et les contrats intelligents. Le protocole de Cardano utilise Ouroboros, un algorithme de consensus basé sur la preuve d'enjeu (Proof of Stake, PoS), qui améliore l'efficacité énergétique et la sécurité. Cardano est développé en deux couches : la couche de règlement (Cardano Settlement Layer, CSL) pour les transferts de valeur et la couche de calcul (Cardano Computation Layer, CCL) pour les contrats intelligents, permettant une flexibilité et une évolution indépendantes des fonctionnalités. Grâce à sa fondation sur des principes académiques et des revues par les pairs, Cardano aspire à offrir une plateforme robuste et fiable pour les systèmes financiers de nouvelle génération.

6. Evolution de la blockchain

Alors que le monde passait par l'un des plus grands effondrements économiques de cette dernière décennie causée principalement par les banques américaines en 2008, cette crise a provoqué un problème de confiance vis-à-vis des intermédiaires et plus précisément des institutions bancaires. Ces intermédiaires dictent leurs lois, abusent parfois de leurs autorités et plus important, concentrent les risques.
C'est dans ce climat de méfiance qu'une personne (ou un groupe de personnes) sous le nom de "Satoshi Nakamoto" a publié un livre blanc ainsi qu'un code de base pour un nouveau type de système monétaire basé sur les travaux des chercheurs Stuart Haber et W. Scott Stornetta de 1991, qui parle principalement de comment horodater un document numérique[100].
Le travail de Nakamoto S. a permis d'introduire une nouvelle méthode pour créer et stocker de la valeur sous forme digitale complètement décentralisée reposant sur une approche de "chaîne de blocs" que nous avons nommée Blockchain. Par la suite, plusieurs domaines autres que le domaine financier tel que les chaînes d'approvisionnement, la vérification des

documents éducatifs [123], le vote, les droits d'auteurs [124] et bien d'autres ont essayé d'intégrer cette technologie.
Le reste de cette section présentera une étude sur l'évolution de la Blockchain et de ses quatre générations. Nous commencerons par la Blockchain 1.0, qui est principalement liée au Bitcoin et aux Crypto-monnaies. puis nous passerons à l'enregistrement et au transfert des contrats intelligents, à savoir Blockchain 2.0. Ensuite nous aborderons l'extension de la Blockchain à divers domaines tels que la gouvernance et l'éducation, qui est indiqué comme Blockchain 3.0. Enfin nous terminerons par voir l'utilisation de la Blockchain dans le secteur industriel en tant que révolution, qui est vue comme Blockchain 4.0.

6.1. Blockchain 1.0: Les cryptos-monnaies

La première génération était entièrement consacrée à la décentralisation de la monnaie et des paiements, bien qu'il s'agisse de la première mise en œuvre d'une technologie de grand registre distribué (DLT pour distributed ledger technology en Anglais). Elle soutient l'exploitation des bitcoins. Le réseau est de type peer to peer et les transactions ont lieu entre les utilisateurs directement sans l'intervention d'une partie tierce. D'autres crypto-monnaies avaient vu le jour telles que Litecoin, Dogecoin, etc. La technologie de bitcoin se repose sur la Blockchain et sur un protocole qui est utilisé pour décrire la manière dont les bitcoins sont transférés. L'algorithme de consensus utilisé est la Preuve du travail (PoW). La Blockchain 1.0 garantit un stockage distribué, permet le partage des données entre les nœuds et assure la transparence des transactions[125].

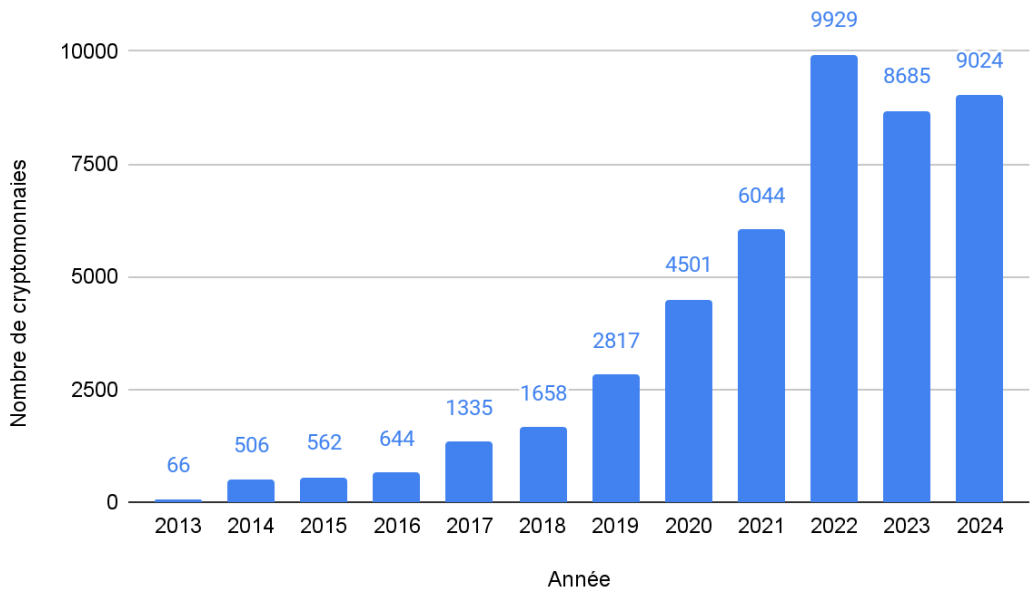


Figure 4.3. Le nombre de crypto-monnaies dans le monde[126].

Le graphique dans la Figure 4.3 montre l'évolution du nombre de crypto-monnaies dans le monde de 2013 à janvier 2024, avec des données provenant de Statista [126]. On peut observer une croissance exponentielle du nombre de crypto-monnaies au cours de la période étudiée. En 2013, il n'y avait que 66 cryptomonnaies. En janvier 2024, ce nombre est passé à

12 500. La croissance a été particulièrement forte en 2017 et 2018, années au cours desquelles le nombre de crypto-monnaies a doublé, voire triplé.

6.2. Blockchain 2.0 : Les contrats intelligents

Dans Blockchain 2.0, un niveau logique a été ajouté dans le ledger qui supporte ce que l'on appelle les contrats intelligents. Les contrats intelligents sont de petits programmes informatiques qui s'exécutent automatiquement lorsque certaines conditions sont remplies. Comme les contrats intelligents sont par essence inviolables, ils réduisent le coût de la vérification, de l'exécution et de la prévention des fraudes. Le concept de contrat intelligent a été proposé pour la première fois par Nick Szabo[127] en 1994 et a été mis en œuvre pour la première fois dans Ethereum dont la proposition a été faite en 2013. La première version de sa blockchain est sortie en juillet 2015. Cette version permet la création et le transfert de biens numériques[125].
Les contrats intelligents [128] sont déployés dans la blockchain sous la forme d'un accord numérique entre deux ou plusieurs autres parties.. Les contrats intelligents sont des programmes informatiques qui peuvent être exécutés de manière cohérente par un réseau de nœuds se méfiant mutuellement, sans l'arbitrage d'une autorité de confiance. Intégrés dans les blockchains, les smart contracts permettent d'appliquer automatiquement les termes contractuels d'un accord sans l'intervention d'un tiers de confiance. Sur la base de sa fonction prédéfinie, il peut stocker, traiter des informations et écrire des résultats. Pour éviter toute falsification, les contrats intelligents sont copiés sur chaque nœud de la blockchain. [129].
La Table 4.3 compare Ethereum, Fabric, Corda sous les aspects suivants : environnement d'exécution, langage de support , protocoles de consensus et le type de la blockchain[130] .

Table 4.3 : Comparaison des plateformes de contrats intelligents

Plateforme	Langage	Protocoles de consensus	Permission
Ethereum	Solidity, Serpent, LLL, Mutan	PoW	Sans(Publique)
Corda	Java, Kotlin	Raft	Avec (Privée)
Fabric	Java, Golang	PBFT	Avec (Privée)

6.3. Blockchain 3.0 : La Convergence vers les DApp (applications décentralisées)

Après les premiers succès de Blockchain 1.0 et 2.0, plusieurs limites ont été révélées. Les plus importantes sont :

- La consommation d'énergie : Comme le minage nécessite une énergie importante (électricité) coûtant des milliards de dollars par an.
- Volume des transactions : Le nombre de transactions augmente toutes les 10-12 secondes à chaque nouvelle création de bloc. Bitcoin peut théoriquement traiter 7 transactions par seconde, tandis que Ethereum en traite 15 par seconde. Si nous comparons le nombre de transactions au réseau Visa, qui traite 24 000 transactions par seconde, nous devons encore améliorer le volume des transactions.

- Coût : étant donné qu'une petite redevance est exigée des mineurs pour la tenue du grand livre, ce système ne convient qu'à un nombre limité de grandes transactions, mais pas aux micro-transactions, car son coût deviendrait prohibitif

Afin de remédier aux limitations des deux premières générations, une troisième génération de Blockchain est actuellement en cours de développement, comme Dfinity, NEO, IOTA et Ethereum, en utilisant différentes approches; ces dernières visent à prendre en charge plusieurs langages de programmation et le développement de diverses applications mobiles[127].

6.4. Blockchain 4.0 : L'intégration transparente avec l'industrie 4.0

Blockchain 4.0 est la dernière génération de la technologie Blockchain. Elle promet de fournir une Blockchain en tant qu'environnement utilisable par les entreprises pour créer et exécuter des applications, ce qui permettra à la technologie de se généraliser [131].
Effectivement la Blockchain jusqu'ici n'avait pas encore surmonté plusieurs obstacles majeurs : la vitesse était beaucoup trop faible et seul un petit nombre de personnes possédait les compétences spécialisées requises pour pouvoir développer des applications décentralisées
La quatrième génération prend tous les avantages des trois précédentes générations et promet des applications décentralisées comparables aux applications centralisées.

7. Les avantages et les défis de la technologie Blockchain

7.1. Services offerts par la Blockchain

Intégrité : Pour des raisons de sécurité, ce programme a été conçu de telle sorte que tout bloc ou même une transaction qui s'ajoute à la chaîne ne peut être modifié, ce qui assure en fin de compte un très haut niveau de sécurité.

Traçabilité : Le format de la Blockchain est conçu de telle sorte qu'il permet de localiser facilement tout problème et de le corriger s'il y en a un. Il crée également une piste d'audit irréversible.

Sécurité : La technologie Blockchain est hautement sécurisée car chaque personne qui entre dans le réseau Blockchain se voit attribuer une identité unique liée à son compte. Cela permet de s'assurer que le propriétaire du compte effectue lui-même les transactions. Le cryptage par bloc dans la chaîne rend plus difficile pour tout pirate informatique de perturber la configuration traditionnelle de la chaîne.

Traitement plus rapide : Avant l'invention de la Blockchain, l'organisation bancaire traditionnelle prenait beaucoup de temps pour traiter et lancer la transaction, mais après l'apparition de la technologie de la Blockchain, la vitesse de la transaction a augmenté dans une très large mesure. Auparavant, le processus bancaire global prenait environ trois jours pour être réglé, mais après l'introduction de la Blockchain, ce délai a été réduit à près de quelques minutes, voire quelques secondes.

7.2. Les défis de la technologie Blockchain

La technologie Blockchain est confrontée à plusieurs défis qui nécessitent une attention particulière afin de garantir son adoption et son utilisation efficace. Parmi ces défis, la consommation d'énergie élevée, les coûts associés et le statut réglementaire incertain dans certains contextes sont particulièrement préoccupants. De plus, il est à noter que les modèles de consensus actuellement disponibles ne sont pas suffisamment efficaces en termes d'évolutivité, ce qui limite leur capacité à offrir une qualité de service optimale (en termes de débit et de latence) pour les applications industrielles pratiques.

1. Consommation d'énergie

La consommation énergétique élevée de la Blockchain dépend en grande partie du protocole de consensus utilisé. Certains protocoles, tels que la preuve de travail (Proof of Work) utilisé par Bitcoin, sont connus pour leur forte consommation d'énergie, tandis que d'autres protocoles, comme la preuve d'enjeu (Proof of Stake) envisagée dans Ethereum 2.0, visent à réduire considérablement cette consommation.

2. Coût

En plus de la consommation d'énergie, les coûts associés aux transactions sur la Blockchain peuvent également poser des défis. Selon certaines études, le coût moyen d'une transaction Bitcoin peut atteindre jusqu'à 160 dollars, principalement en raison de la consommation d'énergie.
Lorsque une transaction est ajoutée à un bloc, elle y prend donc une certaine place en octets. Et comme un bloc est de taille limitée, il est commun qu'il y ait plus de transactions en attente qu'il n'est possible d'en ajouter dans le prochain bloc.
Le mineur qui compose ce prochain bloc a donc la liberté de sélectionner et choisir quelles transactions il souhaite y inclure. En général, les mineurs cherchent à maximiser les frais qu'ils collectent dans le bloc, et vont donc choisir les transactions aux frais les plus élevés. C'est ce mécanisme qui explique que, sauf exception, plus les frais d'une transaction sont importants et plus vite elle sera confirmée par un mineur.
Cette problématique souligne l'importance de trouver des solutions durables pour réduire les coûts tout en maintenant l'efficacité du système.

3. Le statut réglementaire

Le statut réglementaire de la blockchain reste incertain dans de nombreux pays, ce qui constitue un obstacle majeur à son adoption généralisée. Plusieurs facteurs contribuent à cette incertitude, notamment l'absence d'un cadre juridique clair, la classification variée de la blockchain selon ses applications, et les préoccupations des régulateurs concernant les risques potentiels associés à cette technologie.
Néanmoins, cette incertitude réglementaire freine l'investissement des entreprises dans la blockchain, limitant ainsi le développement de nouvelles applications et services basés sur

cette technologie. Il est crucial que des réglementations claires et transparentes soient établies pour favoriser une adoption généralisée et pérenne de la blockchain dans le futur.

4. Scalabilité

Plusieurs études ont examiné le concept du trilemme de la scalabilité pour la blockchain [132,133]. Initialement, ce concept a été décrit par Vitalik Buterin, le cofondateur d'Ethereum, qui a déclaré que des compromis sont inévitables entre trois propriétés importantes de la blockchain : la décentralisation, l'évolutivité et la sécurité comme illustrées dans la Figure 4.4 La décentralisation est le cœur et la nature de la blockchain. La sécurité est une propriété essentielle, tandis que l'évolutivité est le principal défi. En d'autres termes, le trilemme de l'évolutivité indique que des compromis sont presque inévitables entre ces caractéristiques de la blockchain [134]. Ces propriétés sont corrélées négativement. Ainsi en se concentrant trop sur deux d'entre elles, la troisième sera affectée négativement.
La scalabilité hors chaîne fait référence aux approches qui permettent l'exécution des transactions sans encombrer la blockchain. Les protocoles qui se branchent sur la chaîne permettent aux utilisateurs d'envoyer et de recevoir des fonds, sans que les transactions apparaissent sur la chaîne principale [135].
La scalabilité demeure un défi crucial pour des blockchains telles que Bitcoin et Ethereum, confrontées à des problèmes de débit limité, de latence élevée des transactions et de consommation d'énergie considérable. Cela a conduit à la recherche de solutions hors chaîne pour atteindre la scalabilité tout en préservant la sécurité et la décentralisation de la blockchain principale.

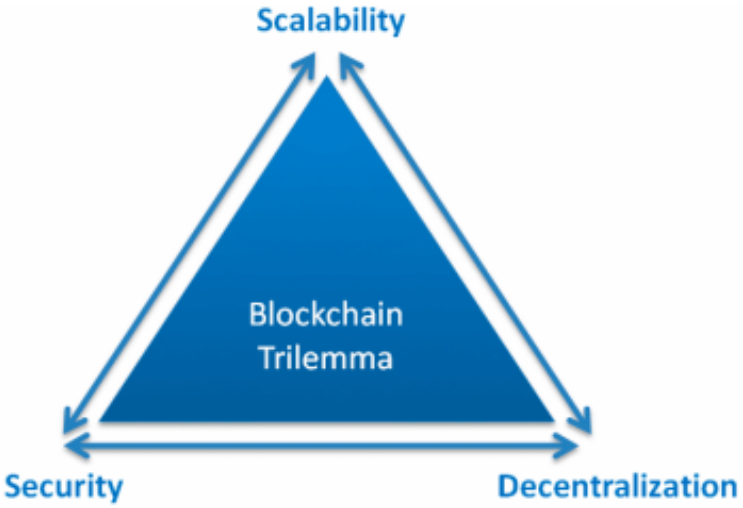


Figure 4.4. Le trilemme de la blockchain[135]

Une approche prometteuse est celle des sidechains (chaînes latérales), qui permettent l'exécution des transactions sans surcharger la blockchain principale. Les sidechains sont des blockchains distinctes, interopérables avec la chaîne principale, permettant ainsi le transfert d'actifs entre les deux. Ces sidechains offrent la possibilité d'expérimenter de nouveaux mécanismes de consensus et d'implémenter des fonctionnalités qui ne seraient pas réalisables sur la chaîne principale. Elles peuvent ainsi contribuer de manière significative à la scalabilité

de la blockchain tout en offrant des avantages supplémentaires en termes de rapidité et d'efficacité des transactions.
Une sidechain est une blockchain distincte. Toutefois, il ne s'agit pas d'une plateforme autonome, car elle est rattachée d'une manière ou d'une autre à la chaîne principale. La chaîne principale et la sidechain sont interopérables, ce qui signifie que les actifs peuvent circuler librement de l'une à l'autre.
Les sidechains fonctionnent selon des règles différentes de celles de la chaîne principale. En réalité, elles ne sont même pas tenues d'utiliser la preuve de travail pour leur fonctionnement. Elles offrent la flexibilité d'utiliser divers mécanismes de consensus, de faire confiance à un seul validateur ou de modifier plusieurs paramètres. Les sidechains permettent d'apporter des améliorations inexistantes sur la chaîne principale, telles que la production de blocs plus importants et l'application de règlements plus rapides.
Un aspect remarquable des sidechains est leur capacité à présenter des bogues critiques sans affecter la chaîne principale. Cela en fait des plateformes idéales pour l'expérimentation et le déploiement de fonctionnalités qui nécessitent normalement le consensus de la majorité du réseau.

5. Attaques classiques sur une blockchain

Les blockchains sont considérées comme des systèmes de stockage de données fiables et sécurisés, mais cela ne signifie pas qu'elles sont à l'abri des attaques. La Blockchain est concernée par les attaques classiques sur les données , nous nous contentons dans cette section de présenter les attaques spécifiques aux blockchains :
Attaque 51 % : elle peut s'appliquer à toute blockchain basée sur l'algorithme de consensus PoW, où la puissance de calcul est utilisée pour résoudre des problèmes complexes et valider les transactions. Elle se produit lorsqu'un attaquant ou un groupe d'attaquants contrôle 51 % ou plus de la puissance de calcul du réseau de la blockchain et qu'il est donc en mesure d'influencer le consensus de la blockchain, d'annuler des transactions antérieures, de miner des blocs et de dépenser plusieurs fois les mêmes fonds. En effet, des attaques à 51 % ont été observées sur plusieurs blockchains, en particulier sur les petits réseaux et les altcoins qui ont une faible participation et une faible puissance de calcul. La protection contre ce type d'attaque se fait en favorisant la décentralisation, en encourageant la participation de nombreux mineurs et en diversifiant la puissance de calcul sur le réseau [136].
Attaque par réplication : il s'agit d'une menace par laquelle des copies de la blockchain légitime sont créées pour introduire des transactions frauduleuses ou modifier l'historique des transactions. Cela peut compromettre l'intégrité de la blockchain. Les utilisateurs peuvent prévenir ce type d'attaque en veillant à utiliser des algorithmes de consensus robustes qui garantissent la validité des blocs et en vérifiant l'intégrité de la chaîne en comparant les versions entre les nœuds du réseau [137]
Le minage égoïste : il s'agit d'une attaque par laquelle un mineur malveillant garde pour lui les blocs nouvellement minés au lieu de les partager avec le reste du réseau. Cela lui donne un avantage injuste en termes de récompenses minières et perturbe l'équilibre du système. En effet, pour éviter cette attaque, il est important d'utiliser des protocoles de consensus

équitables qui récompensent une participation honnête et de surveiller attentivement l'activité des mineurs pour détecter tout comportement suspect [138].

8. Les Applications de la Blockchain

La Blockchain peut être utilisée dans les différents domaines industriels et techniques. Les plus grandes sociétés informatiques mettent en œuvre cette technologie pour améliorer la qualité et la capacité de travail des systèmes. La liste des applications déjà existantes n'est pas finie et nous pensons que d'autres applications Blockchain apparaîtront dans un avenir proche dans des domaines aussi divers que l'art, le tourisme, le sport et autres. Bien qu'encore à leurs débuts, il ne faut pas sous-estimer les avantages socio-économiques prometteurs de ces changements technologiques extraordinaires [139].
Dans cette section, nous présentons certaines applications de la technologie Blockchain.

8.1. Les systèmes de vote basés sur la Blockchain

En février 2015, la Fondation Bitcoin (2015) a dévoilé un nouveau projet qui s'articule autour d'un système de vote basé sur la blockchain, qui offre une transparence encore plus grande dans le processus de vote, chaque vote étant enregistré sur la blockchain s'appuyant sur l'immuabilité, la transparence et le consensus inhérents à la technologie des blockchains, les systèmes de vote, où chaque vote est enregistré sous un code hash cryptographique sécurisé, apparaissent comme une avancée technologique majeure. À la jonction entre la démocratie électronique et la technologie de la blockchain, ce type de système de vote a d'abord été mis en œuvre par un parti politique danois pour des élections internes[140].

8.2. Améliorer la transparence des chaînes d'approvisionnements:

Pour accroître l'efficacité du fret maritime, Maersk le plus grand armateur de porte-conteneurs du monde et IBM ont lancé une initiative visant à établir un système mondial basé sur la blockchain pour numériser les flux commerciaux et le suivi des expéditions de bout en bout. Le système permet à chaque partie prenante de la chaîne d'approvisionnement de visualiser la progression des marchandises tout au long de la chaîne, en comprenant où un conteneur est en transit. Les parties prenantes peuvent également voir le statut des documents douaniers et peuvent consulter les connaissements et d'autres données. La technologie de la Blockchain garantit un échange de données sécurisé et un dépôt inviolable de cette documentation [141].

8.3. L'identité numérique

La blockchain pourrait être utilisée autant que moyen sûr de vérification d'authenticité d'une personne, cela implique la réduction ou même l'élimination des possibilités de fraudes. L'Estonie par exemple a mis en œuvre le système E-Residence comme système d'identification électronique basée sur la technologie blockchain pour ses citoyens [142].

8.4. La blockchain pour M2M/ IoT

L'intégration de la technologie blockchain dans les réseaux Machine à Machine (M2M) offre des solutions pour renforcer la sécurité, la transparence et l'efficacité de ces systèmes. La blockchain garantit une sécurité renforcée grâce à son caractère immuable et son cryptage robuste, ce qui rend les systèmes M2M plus résistants aux cyberattaques. De plus, elle permet une traçabilité complète des données, réduisant ainsi les risques de fraude. La décentralisation du contrôle grâce à la blockchain donne aux utilisateurs et aux appareils M2M plus d'autonomie et de contrôle sur leurs données, renforçant ainsi la confidentialité.
Sur le plan opérationnel, la blockchain facilite l'automatisation des processus et les transactions sécurisées entre les appareils et les systèmes, ce qui peut entraîner une amélioration significative de l'efficacité opérationnelle.
Enfin, l'intégration de la blockchain ouvre de nouvelles perspectives commerciales en permettant la création de marchés décentralisés pour les données M2M, offrant ainsi de nouvelles sources de revenus. Cependant, des défis subsistent, notamment en termes d'évolutivité des blockchains publiques pour les réseaux M2M à grande échelle, de consommation énergétique des technologies de consensus blockchain et de complexité liée à l'intégration de la blockchain dans les systèmes M2M[135]. Cependant, les attaques et les défaillances de sécurité pourraient causer d'énormes maux de tête aux réseaux M2M. Par exemple, les centres de données centraux sont vulnérables aux défaillances ponctuelles et aux attaques malveillantes. En outre, la communication entre les dispositifs peut faire l'objet d'une interception des données, et la crédibilité des données collectées ne peut être garantie.

8 4.1. Blockchain pour la sécurité des communications M2M /IoT

Ces dernières années, avec l'émergence de la blockchain, l'idée de combiner la blockchain avec l'internet des objets a suscité une grande attention [143,144]. En tirant parti des caractéristiques du mécanisme de consensus décentralisé et inviolable de la blockchain, il est possible de résoudre les problèmes de sécurité des systèmes de l'IIoT décrits ci-dessus. Il existe quelques recherches sur ce sujet, par exemple, O. Novo [143] propose un système de contrôle d'accès basé sur la technologie blockchain pour gérer les appareils IoT. Cependant, le système n'est pas entièrement construit sur une architecture distribuée en raison de l'utilisation d'un centre de gestion central. Lorsque le centre de gestion est défaillant ou attaqué, les appareils IoT qui y sont connectés deviennent indisponibles. Z. Li et al [145] exploitent la technologie blockchain de type consortium pour proposer un système d'échange d'énergie sécurisé. Mais ils ne prennent pas en compte les questions de confidentialité, telles que le risque de divulgation des données sensibles, et ne peuvent donc pas garantir la sécurité des données sensibles.
Les systèmes susmentionnés adoptent tous des blockchains structurées en chaîne dans les systèmes IoT, qui sont surchargées pour les dispositifs IoT à puissance limitée. Z. Xiong et al [146] introduisent l'informatique périphérique (Edge computing) pour les applications blockchain mobiles et présentent un modèle de jeu de Stackelberg pour une gestion efficace des ressources périphériques pour la blockchain mobile. Ils réduisent les exigences de calcul des appareils mobiles en tirant parti de l'informatique périphérique. En outre, d'autres défis

sont apparus lors de l'introduction de la nouvelle conception de la blockchain dans les systèmes IIoT.
Les auteurs de [147] utilisent une approche basée sur la blockchain pour sécuriser le processus de routage en considérant la blockchain comme une mémoire partagée dans le WSN. Afin d'optimiser et de sécuriser la phase de routage, la blockchain conserve la trace des transactions générées sur le réseau, indiquant quels nœuds transmettent et par quels chemins.
Un processus de sécurisation des informations basé sur la blockchain, dans lequel les données sont collectées à partir des capteurs en utilisant un véhicule aérien sans pilote (UAV) comme relais, est présenté dans [148]. Les informations collectées sont stockées en toute sécurité dans la blockchain sur le serveur informatique mobile (MEC pour mobile edge computing). Dans le cadre du système proposé, les informations sont cryptées avant d'être transférées au serveur MEC à l'aide du drone. Lorsqu'il reçoit l'information, le serveur MEC approuve l'information et l'identité de l'expéditeur. La validation effective est suivie du stockage de l'information dans la blockchain.
Dans [149], les auteurs présentent un système basé sur la blockchain en combinaison avec un mécanisme de chiffrement par procuration pour garantir la confidentialité des informations. Les transactions correspondantes sont donc gérées par le contrat intelligent convenu et sauvegardées dans la blockchain.
Une architecture de blockchain décentralisée dans [150] est adoptée dans le but principal de faciliter l'échange d'informations sécurisées au sein d'une APL (plateforme de protection adaptative). L'architecture augmente le débit tout en conduisant à une meilleure évolutivité et à un faible stockage de nœuds.
Une approche de gestion d'identité légère est présentée dans [151]. Elle est basée sur une blockchain de consortium pour l'IdO. Le protocole aborde les problèmes de confidentialité et de sécurité dans les systèmes centralisés traditionnels et se concentre sur le concept de dispersion de l'autorité d'un système d'identité à un groupe d'organisations. La preuve de concept (PoC) en tant que modèle de consensus modérément léger est mise en œuvre.
Dans [152], le modèle proposé fournit une zone d'observation sécurisée pour vérifier les données IoT générées par les appareils IoT. À l'aide du protocole MAM (Masked Authentication Messaging), les données cryptées des capteurs environnementaux sont diffusées, stockées et extraites d'un grand livre distribué pour l'IdO, appelé IOTA, afin de garantir l'intégrité, la sécurité et la confidentialité.

9. Conclusion

La Blockchain représente une avancée significative dans le domaine de la technologie de l'information, offrant un cadre transparent, sécurisé et décentralisé pour la gestion des transactions et des données. Ce chapitre a exploré en détail les concepts fondamentaux de la Blockchain, y compris ses protocoles de consensus, ses différents types, et son évolution à travers les différentes générations. Nous avons également examiné ses avantages, tels que la sécurité accrue et la réduction des coûts de transaction, ainsi que ses défis, notamment la consommation d'énergie élevée, le statut réglementaire incertain et les limitations en termes de scalabilité.

Malgré ces défis, la Blockchain continue d'être explorée et adoptée dans divers secteurs, notamment pour améliorer la sécurité des communications IoT/M2M. En combinant des approches innovantes avec la Blockchain, de nouvelles possibilités émergent pour renforcer la confidentialité, l'intégrité et la résilience des communications dans un environnement de plus en plus connecté. En l'occurrence, dans le cadre de cette thèse, l'intégration de la technologie Blockchain pour assurer la sécurité des communications M2M a fait l'objet de nos recherches. Le chapitre suivant portera sur les architectures et mécanismes basés Blockchain, que nous proposons pour sécuriser les communications dans le domaine des réseaux de dispositifs M2M sans fil.

Chapitre 5

APPROCHES POUR LA SÉCURITÉ DES COMMUNICATIONS M2M

1. Introduction

Dans les chapitres précédents, nous nous sommes intéressés à différents aspects de l'architecture M2M (caractéristiques, contraintes,…), insistant en particulier sur le volet de la sécurité des communications M2M (Machine à Machine). Notre étude de l'état de l'art, nous a permis de comprendre les nuances des diverses couches de communication M2M, les protocoles en place, ainsi que les architectures et solutions de sécurité existantes. Les nouvelles tendances pour la sécurisation des réseaux des dispositifs M2M nous ont amené à explorer le domaine de la blockchain et ses potentialités à répondre à notre problématique, examinant ses concepts fondamentaux, les protocoles de consensus, les types et les applications dans le contexte de l'IoT/M2M. Les solutions, que nous avons proposées dans le cadre de cette thèse, s'articulent autour d'une architecture basée sur une blockchain placée sur une infrastructure Edge, à la périphérie du champ des réseaux des dispositifs M2M (M2M Device Area Network) .
Dans ce chapitre, consacré à nos contributions, nous présentons les résultats de nos travaux que nous avons soumis sous formes d'articles scientifiques distincts pour publications dans des journaux et conférences internationales.

1) Mécanisme d'authentification léger basé sur la blockchain pour les communications M2M dans l'IoT (Lightweight Blockchain-Based Scheme to Secure Wireless M2M Area Networks).
Cette contribution propose un schéma léger basé sur une blockchain privée pour sécuriser les communications sans fil M2M au niveau du domaine réseau des dispositifs. Il offre un stockage sécurisé des données tout en préservant l'intégrité, la traçabilité et la disponibilité des données échangées.

2) Mécanisme d'authentification léger basé sur une blockchain à deux couches pour les communications M2M dans l'IoT (Blockchain-based Authentication Protocol for Wireless M2M Area Networks with Sidechain Integration)

Alors que la première contribution proposait un mécanisme d'authentification basé sur une blockchain privée pour sécuriser les communications M2M au niveau du domaine des dispositifs M2M, ses résultats ont mis en lumière des défis de scalabilité inhérents aux solutions basées sur une seule blockchain. En effet, les dispositifs M2M, caractérisés par leurs ressources limitées, peuvent rencontrer des difficultés à gérer les exigences computationnelles des blockchains traditionnelles. Pour aborder cette problématique, le deuxième article de cette thèse présente une approche reposant sur l'intégration de deux blockchains (mainchain et sidechain) au sein d'un réseau de stations de base situé au niveau de la périphérie, proche du champ de détection M2M.

3) Protocole de communication multi-sauts sécurisé et économe en énergie (SEEM-D2D: Secure and Energy Efficient Multi-hop D2D Communications in Wireless M2M Area Networks using Two-Layer Blockchain)

Cette contribution propose un protocole de communication multi-sauts sécurisé et économe en énergie pour les réseaux M2M sans fil, utilisant une blockchain à deux couches avec intégration d'une sidechain (chaîne latérale). Il garantit une authentification préliminaire sécurisée et permet des communications efficaces entre machines via des relais intermédiaires en mode multi-sauts. En outre, il est important de noter que ce protocole permet d'assurer les communications D2D sans avoir recours à une station de base, ce qui renforce encore plus l'efficacité et la flexibilité du système.

2. Contributions

2.1. Mécanisme d'authentification léger basé sur la blockchain pour les communications M2M dans l'IoT (*Lightweight Blockchain-Based Scheme to Secure Wireless M2M Area Networks*).

2.1.1. Introduction

La sécurité représente un défi majeur pour les applications M2M/IoT en raison du déploiement, de la décentralisation et de l'hétérogénéité des dispositifs M2M et IoT. Les solutions de sécurité typiques peuvent ne pas être adaptées aux systèmes M2M/IoT en raison des difficultés rencontrées pour leur mise en œuvre sur des dispositifs aux ressources limitées. Dans ce contexte, nous proposons un mécanisme d'authentification et d'autorisation léger basé sur une blockchain privée afin de sécuriser les communications sans fil M2M au niveau du domaine réseau des dispositifs. L'intégration de la blockchain offre un stockage sécurisé des données tout en préservant l'intégrité, la traçabilité et la disponibilité des informations échangées. Nous présentons par la suite un état de l'art sur différents protocoles de sécurité pour les communications M2M sans fil, suivie d'une description détaillée de l'architecture de communication M2M standard et de ses différents composants. Ensuite, nous exposons en détail le schéma d'authentification proposé pour le domaine des dispositifs M2M, en mettant en lumière ses phases de pré-enregistrement, d'enregistrement et d'authentification. Nous analysons également les performances de ce schéma à l'aide de simulations expérimentales et nous comparons ses résultats avec d'autres protocoles similaires. Enfin, nous discutons des avantages introduits par notre approche et des directions de recherche futures envisageables.

2.1.2. Architecture du domaine des dispositifs M2M

Dans l'architecture proposée pour le domaine des dispositifs M2M (Figure. 5.1), les réseaux de périphériques sont connectés à la couche réseau via des passerelles en fonction des technologies particulières (Wi-Fi, Zigbee, réseaux cellulaires 4G et 5G, etc.). La Table 5.1 présente les paramètres utilisés avec leurs longueurs en bits.

Table 5.1 : Notations et longueurs des paramètres.

Symbol	Description(Anglais)	Description(Français)	Length (bits)
DID	Device identity	Identité de l'appareil	128
DPbK	Device Public Key	Clé publique de l'appareil	160
DPrK	Device Private Key	Clé privée de l'appareil	160
BSPbK	Base Station public Key	Clé publique de la SB	160
BSPrK	Base Station private Key	Clé privée de la station de base	160
PsK	Pre-shared key	Clé pré-partagée	128
SK	Session key	Clé de session	80
MAC	Message Authentication Code	Code d'authentification du message	256
Tstmp	TimeStamp	Horodatage	32

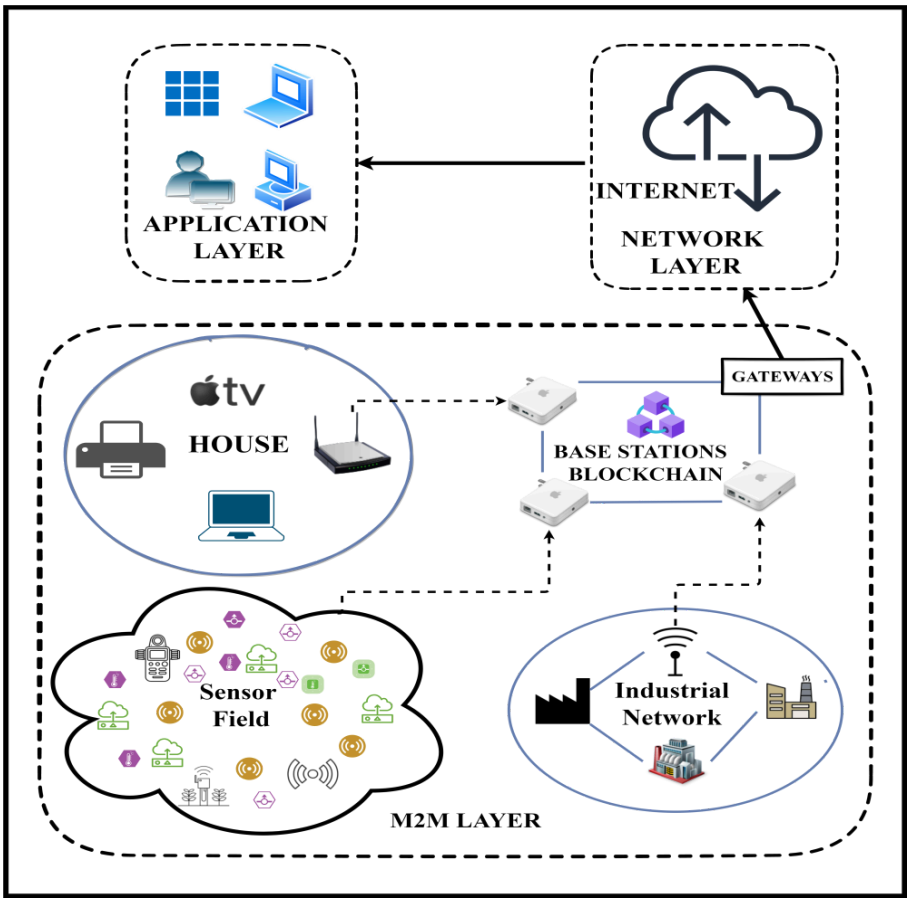


Figure. 5.1. Architecture de la Couche M2M.

Les stations de base incorporent des passerelles, présentées dans la Figure. 5.2 sont supposées avoir des ressources computationnelles et énergétiques élevées (ou illimitées).

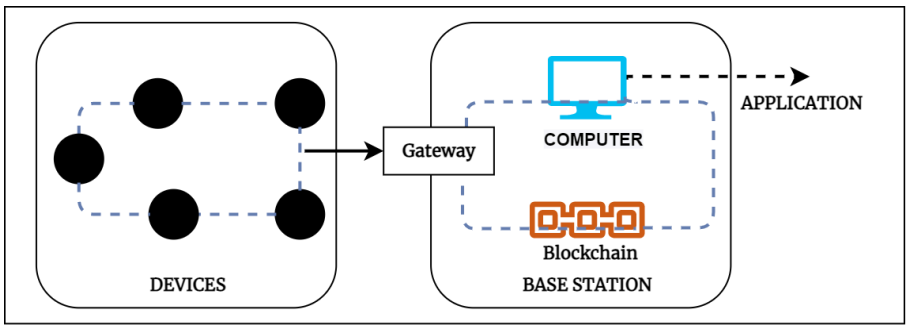


Figure. 5.2. Station de base M2M.

Elles effectuent les tâches suivantes :

- Enregistrement d'un nouveau nœud sur le réseau en tant qu'entité nouvelle.
- Gestion de l'identité et de l'authentification des communications sécurisées.

Les nœuds M2M (capteurs, objets connectés, etc.) collectent et transmettent des paquets de données avant de les transmettre aux passerelles M2M. Après réception des paquets, une passerelle M2M gère les paquets et fournit des chemins appropriés pour les transmettre à un serveur d'application M2M ou à d'autres nœuds de périphériques sur différents réseaux de périphériques via des réseaux câblés/sans fil. Bien entendu, les passerelles effectuent une conversion de protocole appropriée et une segmentation ou assemblage des données (par exemple, de IP à 802.15.4 et vice versa).

2.1.3. Schéma d'authentification proposé

La technologie de la blockchain en tant que système distribué permet d'éviter le Point de Défaillance Unique (SPF) des schémas centralisés. La blockchain nous permet de résoudre les problèmes de sécurité tels que la gestion des identités, la disponibilité, l'intégrité et offre une solution pour la traçabilité afin de suivre l'historique de chaque activité de périphérique tout en enregistrant toutes ses transactions.

Étant donné que les périphériques contraints ne peuvent pas supporter le traitement lourd que la technologie blockchain introduit pour le minage et les opérations cryptographiques, une blockchain privée est adoptée et située sur un ensemble de stations de base plus proches du champ détecté (c'est-à-dire la zone M2M), ce qui présente plusieurs avantages, car cela permet des retards courts et un contrôle précoce des appareils. La blockchain est conçue pour contenir toutes les informations relatives à chaque appareil. La Figure. 5.3 représente le contenu et la structure des blocs.

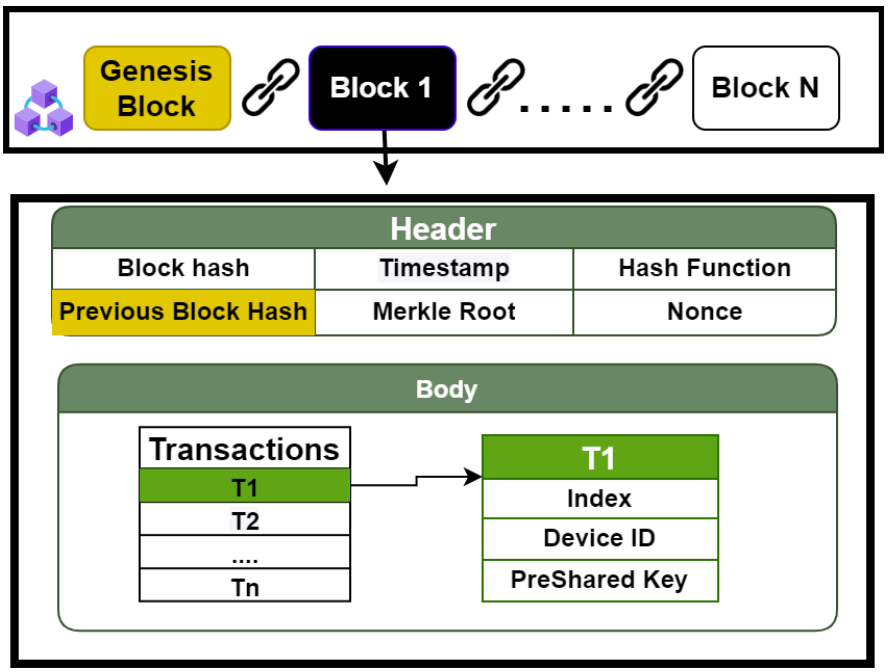


Figure. 5.3. Structure de bloc de la blockchain

D'autre part, nous choisissons Practical Byzantine Fault Tolerance (PBFT) à implémenter comme algorithme de consensus léger adapté à la blockchain privée. Le consensus PBFT garantit l'intégrité et la fiabilité des données même si N/3 -1 des N nœuds du réseau sont compromis. Ainsi, un nouveau bloc est validé et ajouté à la blockchain si (2 ∗ N/3) +1 nœuds, au minimum, atteignent le consensus.
Avant que la communication ait lieu, le processus d'identification est lancé avec trois phases : pré-enregistrement, enregistrement et authentification.

2.1.3.1. Phase de pré-enregistrement

Une clé pré-partagée (PsK) utilisée pour l'enregistrement d'un appareil peut être générée par l'application M2M après vérification de la non-existence de l'appareil dans la blockchain. PsK est envoyée à l'appareil via un canal sécurisé, ou automatiquement stockée dans les appareils lors de la configuration initiale de l'appareil (avant le déploiement) et communiquée au serveur d'application M2M via un canal sécurisé comme illustré dans la Figure. 5.4.
L'identifiant de l'appareil est transféré via un canal sécurisé pour être ajouté dans la blockchain privée en utilisant le contrat intelligent déployé dans la Figure. 5.5. Tout d'abord, il vérifie l'existence de l'appareil dans la blockchain en utilisant la fonction IsRegistered() du contrat intelligent, après quoi le nœud peut être ajouté à la blockchain uniquement s'il n'existe pas déjà en appelant la fonction AddDevice().
L'ordre d'enregistrement doit être validé par les nœuds du réseau grâce au consensus PBFT, afin qu'il puisse être stocké dans la blockchain privée. On suppose que les stations de base sont déjà enregistrées au niveau du réseau.

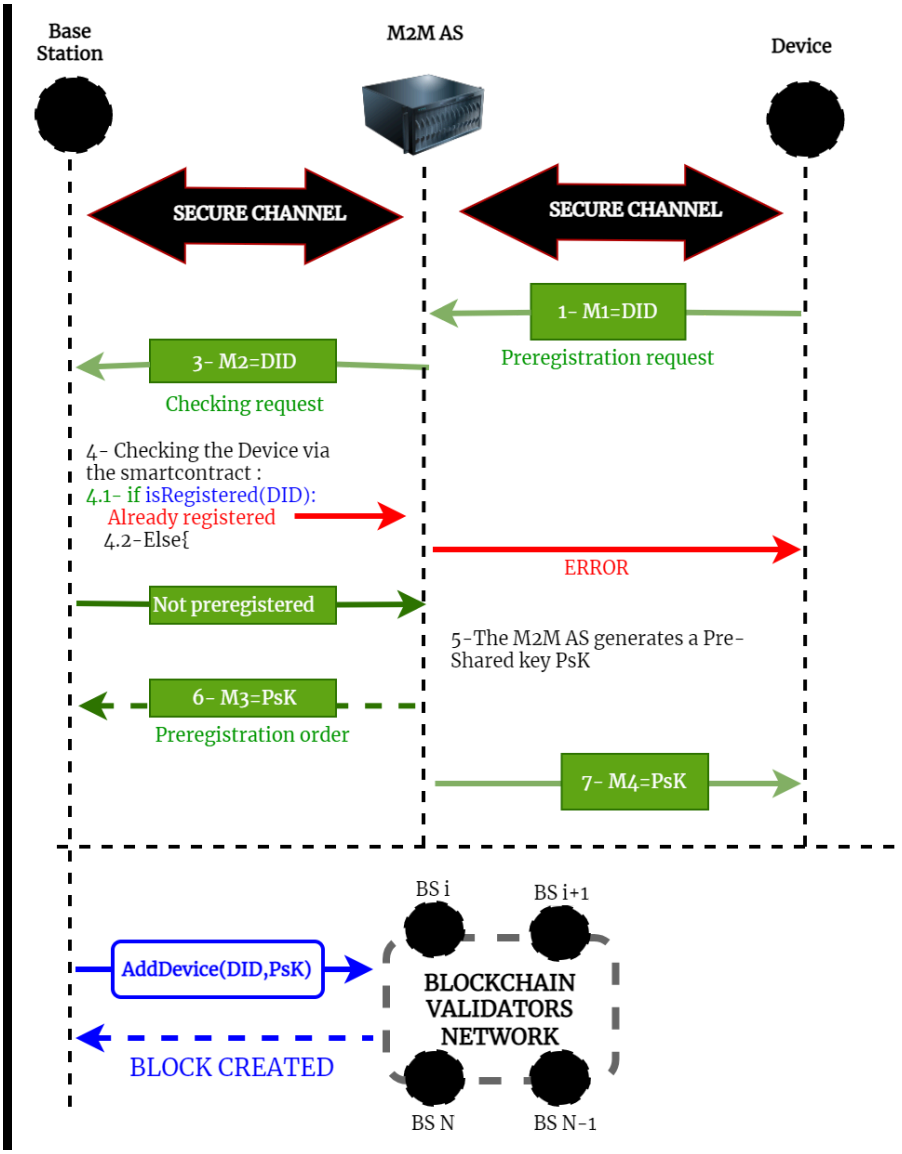


Figure. 5.4. Phase de pré-enregistrement.

Avant de commencer l'échange d'informations, la BS et l'appareil final doivent établir une clé de session (SK) à utiliser pour le chiffrement symétrique. Un AES-128 bits est le chiffrement par bloc le plus efficace (approuvé par le NIST) pour des raisons de confidentialité en termes de sécurité. Cependant, sa mise en œuvre sur des appareils à ressources limitées est inappropriée en raison de ses cycles d'exécution élevés et de sa consommation de mémoire. De plus, une variante dérivée légère d'AES telle que PRESENT [76] avec une clé partagée de 80 bits et un bloc de 64 bits, présentant des performances acceptées, est suggérée pour garantir la confidentialité dans notre conception.

La clé de session est obtenue à partir du point de base de la courbe ECC utilisé par le protocole d'échange de clés ECDH[74] pour être transférée à l'appareil pour un stockage sécurisé. La clé de session a une période de validité et peut être régénérée pour chaque nouvelle session.

Les clés privées DPrK et BSPrK sont des entiers choisis de manière aléatoire dans l'intervalle [1, n - 1], où n est l'ordre de la courbe. Les clés publiques de la station de base et de l'appareil sont corrélées avec leurs clés privées respectives.

Puisque : DPbK = DPrK * G et BSPbK = BSPrK * G où G est le point de base sur la courbe, la clé secrète partagée est alors donnée par l'Équation (1.1) :

$$BSPrK*DPbK=BSPrK(DPrK* G)=DPrK(BSPrK*G)=DprK*BSPrK \quad (1.1)$$

Lorsqu'un appareil sans fil tente de se connecter au réseau pour la première fois, la phase d'enregistrement sera lancée. L'appareil commence par scanner l'environnement et demande la connexion à la station de base la plus proche.

Contract: Authentication Scheme

Initialisation:
- Event AddDev (uint256 index, uint256 DeviceID, uint256 PSK) // The arguments passed are stored in transaction logs once the event emitted, these logs are stored on blockchain.
- struct Device { uint256 DeviceID; string DeviceName;
- string DeviceType; uint256 PublicKey; }
- mapping(uint256 => Device) public allDevices;
- uint256 public index;

Functions

Function 1: isRegistred(uint256){}
Function 2: AddDevice(uint256,uint256){}
Function 3: GetAllTheInfo(uint256){}
Constructor () public { index = 0; }

Function 1: isRegistered

Input : uint256 DeviceID
Result: returns a boolean // checks the existence of a device in the blockchain
If (AllDevices[DeviceID].DeviceID!= 0) { return true; }
else { return false; }

Function 2: AddDevice // allows the addition of a device in the blockchain

Input : uint256 DeviceID, uint256 PSK
Result: a new block is created (the device is added to the blockchain)
emit AddDev(index , DeviceID , PSK);
device = Device(DeviceID, PSK);
AllDevices[DeviceID] = device;
index = index + 1;

Figure. 5.5. Contrat intelligent du schéma d'authentification.

2.1.3.2. Phase d'enregistrement

La phase d'enregistrement se compose de cinq étapes (Figure. 5.6), détaillées comme suit :

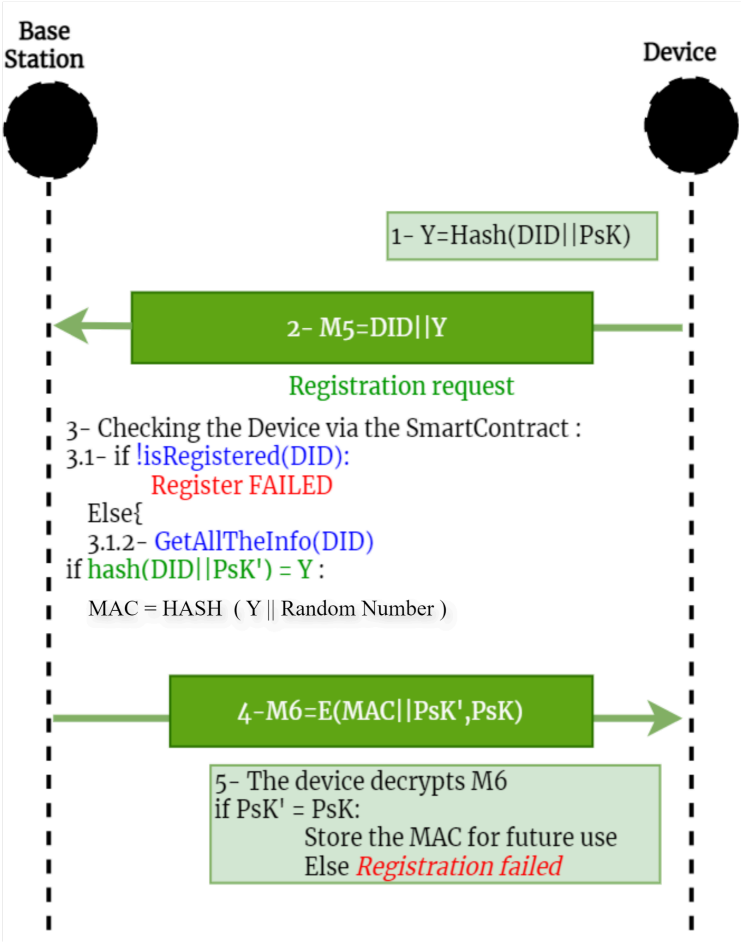


Figure. 5.6. Phase d'enregistrement.

Étape 1 : L'appareil calcule Y, le hash de (DID || PsK). Une variante légère de SHA-3 avec 256 bits est adoptée, comme indiqué par le NIST, la plus susceptible d'être utilisée dans le domaine des conceptions contraintes.
Étape 2 : L'appareil envoie DID et Y à la station de base.
Étape 3 : La station de base vérifie l'existence de l'appareil dans la blockchain, et si elle n'existe pas, l'enregistrement est refusé. Dans l'autre cas, la fonction 3, "GetAllTheInfo", dans la Figure. 5.7 du contrat intelligent est appelée pour obtenir les informations sur l'appareil. La station de base calcule hash(DID || PsK) et le compare à Y, et si la comparaison est vérifiée, elle calcule le MAC (Message Authentication Code) d'une façon pseudo aléatoire comme suit : MAC = Hash(Y || RandomNumber), où Y est le contenu du message reçu .
Étape 4 : La station de base chiffre le MAC et le PsK' en utilisant le PsK où PsK' est la clé pré-partagée de l'appareil stockée dans la blockchain.

Étape 5 : L'appareil déchiffre M6 et vérifie si PsK' est égal à son PsK, le MAC est alors stocké pour une utilisation ultérieure si la comparaison est vérifiée, sinon l'enregistrement échoue.

Function 3: GetAllTheInfo // returns the device information

Input : uint256 DeviceID
Result: Returns uint256, uint256 // the pre shared key and the previous block hash
Return (AllDevices[DeviceID].PSK , block.blockhash(AllowedDevices[DeviceID].index-1));

Figure. 5.7. Fonction GetAllTheInfo().

2.1.3.3. Phase d'authentification

L'authentification de tout appareil se fait avec un code d'authentification de message (MAC) valide car le contrat intelligent est déployé sur toutes les stations de base, ce qui permet de trouver le hachage du bloc précédent contenant l'identifiant de l'appareil. Ce mécanisme renforce la sécurité du réseau.
La phase d'authentification est composée de sept étapes (Figure. 5.8) décrites comme suit :
Étape 1 : L'appareil génère sa clé privée et calcule sa clé publique.
Étape 2 : L'appareil calcule Z où Z = Hachage(MAC || PSK || DPbK).
Étape 3 : L'appareil envoie DID, Z, DPbK à la BS.
Étape 4 : La BS vérifie le timestamp (la valeur du timestamp doit être dans une plage acceptable du temps actuel). La BS vérifie ensuite si l'appareil est déjà enregistré, si ce n'est pas le cas, elle annule l'authentification, sinon, elle vérifie si l'appareil a déjà été authentifié (en vérifiant son existence dans la table des appareils authentifiés), s'il n'existe pas dans la table, la BS collecte les informations sur l'appareil en appelant la fonction GetAllTheInfo(DID) du contrat intelligent, puis elle calcule X où X = DID XOR PH, après cela, elle vérifie si Y est égal à hachage(X || PsK' || DPbK || Tstmp), l'authentification échoue si la comparaison ne tient pas, sinon l'appareil est authentifié avec succès.
Étape 5 : La BS génère sa clé privée, et calcule sa clé publique où BSPbK= BSPrK * G, et calcule la clé de session où SK = BSPrK*DPbK.
Étape 6 : La BS chiffre SK et BSPbK en utilisant la clé publique de l'appareil, le résultat sera envoyé à l'appareil.
Étape 7 : L'appareil déchiffre M4 et calcule la clé de session, le résultat sera comparé à la clé de session reçue dans M4, la BS est authentifiée avec succès si la comparaison est valide, sinon l'authentification échoue.

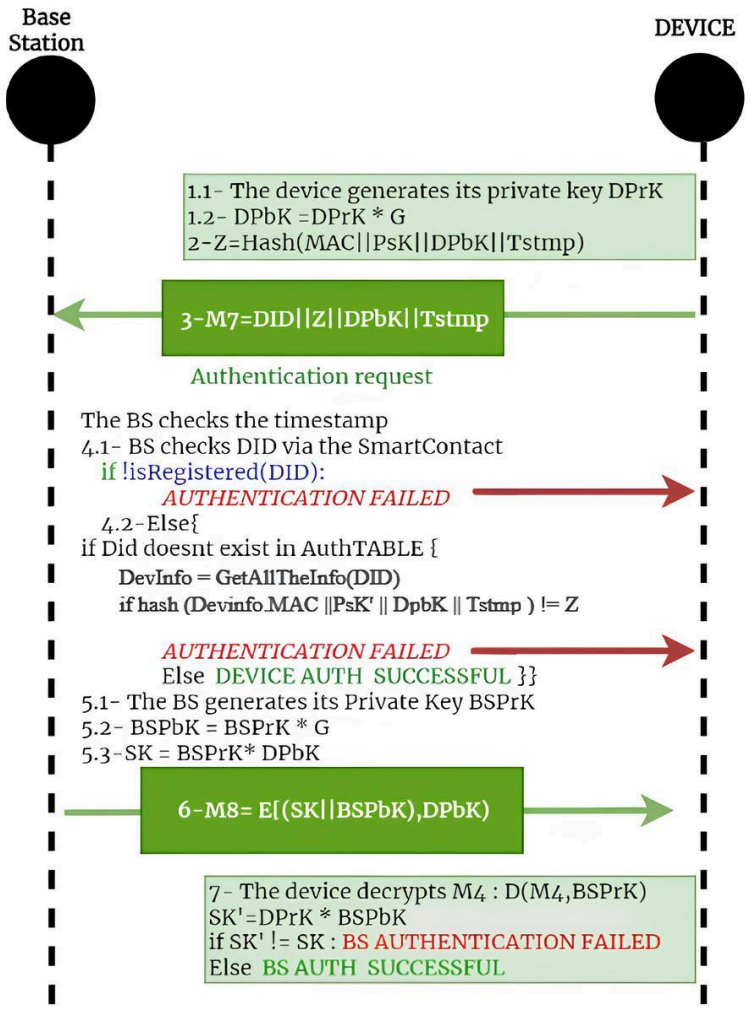


Figure. 5.8. Phase d'authentification.

2.1.4. Évaluation des performances

Dans le cadre de notre analyse comparative des protocoles de sécurité pour les communications M2M, nous avons examiné plusieurs approches traditionnelles ainsi que de nouvelles méthodologies émergentes. Nous avons identifié d'autres protocoles dans la littérature, qui représentent un éventail diversifié de solutions sécurisées pour les réseaux M2M et IoT.

Les auteurs de[153] proposent un cadre d'authentification à trois facteurs basé sur un modèle de publication-abonnement utilisant le protocole de communication d'application de file de messages (MQTT). L'architecture générale prend en charge l'authentification mutuelle des entités entre un utilisateur distant (abonné), un courtier (passerelle) et un nœud IoT (éditeur), basée sur le mot de passe, l'identité et une signature numérique à faible coût, utilisant la cryptographie à courbes elliptiques.

Un schéma de clé authentifié basé sur une signature pour traiter les problèmes de sécurité dans l'IoT est présenté dans[154]. L'authentification mutuelle entre l'utilisateur et le dispositif

final est vérifiée à l'aide de la logique de Burrows–Abadi–Needham (également connue sous le nom de logique BAN ou logique BAN largement acceptée).

Un protocole d'authentification à trois facteurs pour l'internet industriel des objets visant à garantir la confidentialité de l'utilisateur est présenté dans [155]. Le protocole proposé adopte un extracteur flou qui utilise des informations biométriques pour créer des clés robustes, à utiliser en entrée des techniques cryptographiques standard.

Ensuite, nous évaluons les performances du mécanisme d'authentification proposé en termes de paramètres clés tels que les surcharges de communication et de calcul, le délai moyen et la consommation énergétique introduits par le mécanisme d'authentification.

2.1.3.1. Surcharge de communication

La taille des messages d'authentification échangés (surcharge de communication) est un paramètre important dans l'évaluation des performances ; la surcharge de communication est définie comme la somme des longueurs des messages (en bits) transmis et reçus par un appareil dans le processus d'authentification. Ici, le protocole utilise six messages pour réaliser l'authentification. La surcharge de communication est calculée comme suit :

M1 (128 bits) + M4 (128 bits) + M5 (384 bits) + M6 (384 bits) + M7 (576 bits) + M8 (240 bits) (1.2)

Table 5.2 : Coût de communication.

Protocole	Taille des messages échangés (bits)
Protocole proposé	1840
[8]	3152
[81]	2304
[91]	1920
[153]	2560
[154]	2528
[155]	2688

Les surcharges de communication des différents protocoles sont données dans la Table 5.2. Le protocole proposé présente un stockage supplémentaire limité et un coût de communication limité comparativement à d'autres protocoles, comme illustré dans le graphique de la Figure. 5.9.

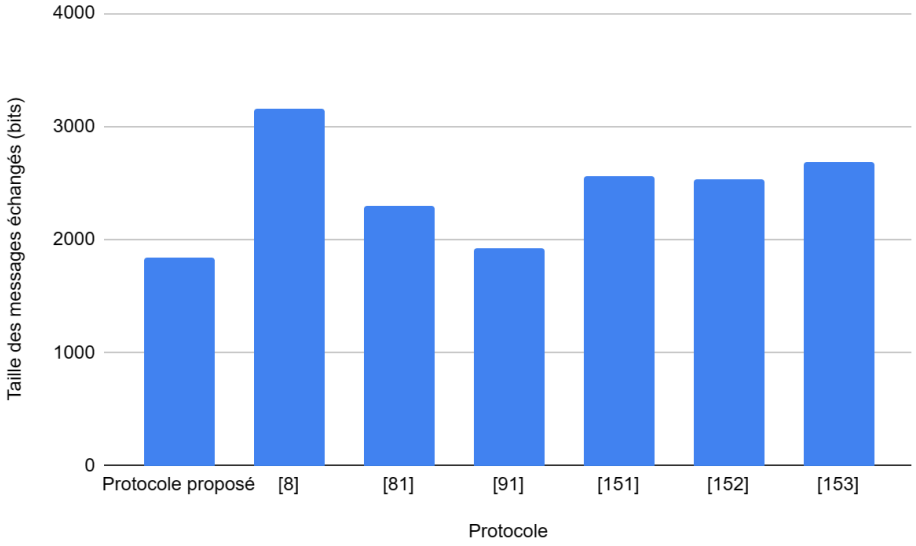


Figure. 5.9. Comparaison des surcharges de communication.

2.1.3.2. La surcharge de calcul

Le coût de la surcharge de calcul prend en considération le temps nécessaire pour effectuer différentes opérations cryptographiques (fonction de hachage, chiffrement et déchiffrement et opérations arithmétiques ECC). L'estimation du coût de calcul est basée sur les mêmes hypothèses utilisées dans [154], comme indiqué dans la Table 5.3.
L'évaluation de la surcharge de calcul a impliqué des mesures détaillées effectuées sur le système d'exploitation Ubuntu 20.04. Nous avons spécifiquement mesuré les opérations cryptographiques pour obtenir des chronométrages précis pour différents processus, ce qui est essentiel pour évaluer l'efficacité computationnelle.

Table 5.3 : Le temps approximatif pour chaque opération cryptographique [154].

Symbole	Fonction	Temps (Sec.)
T (h)	Fonction de hachage	0.00032 s
T (sym)	Chiffrement/déchiffrement symétrique	0.0171 s
T (eca)	Temps d'addition de points ECC	0.0044 s
T (ecm)	Temps de multiplication de points ECC	0.0171 s

Par exemple, les opérations de chiffrement/déchiffrement symétriques, représentées par T(Sym), ont été mesurées à 0.0171s, en utilisant PRESENT [76] comme une variante dérivée légère de AES. Les opérations de chiffrement asymétrique utilisant Diffie-Hellman sur courbe elliptique (ECDH) [74] ont été chronométrées à T(asym)=0.0171s. Les opérations de hachage, représentées par T(hash), ont été quantifiées en utilisant SHA-3 avec 256 bits, donnant un temps de 0.00032s. De plus, l'addition de points ECC, désignée par T(ECA), a été mesurée à 0.0044s.

Les résultats obtenus dans la Table 5.4, illustrés par le graphique dans la Figure. 5.10, montrent que le schéma d'authentification proposé présente le coût total de calcul (TCC) le plus bas, y compris le coût de calcul du dispositif (DCC), bien que le coût de calcul du dispositif soit légèrement plus élevé comparativement à [155].

Table 5.4 : Comparaison du coût de calcul (CC) de l'authentification en secondes.

Protocole	Device CC (DCC)	DCC en Sec.	Total CC (TCC)	TCC en Sec.
Proposé	2 T (h) + 2 T (sym) + T (eca) + T (ecm)	0.0563 s	5 T (h) + 4 T (sym) + T (eca) + 2 T (ecm)	0.113 s
[153]	3 T(ecm) + 1 T (h) + 2 T (eca)	0.0604 s	10 T (ecm) + 7 T (h) + 4 T (eca)	0.190 s
[154]	4 T (ecm) + 3 T (h)	0.0693 s	14 T (ecm) + 12 T (h)	0.243 s
[155]	4 T (h) + 2 T (sym)	0.0354 s	3 T (ecm) + 19 T (h) + 8 T (sym)	0.194 s

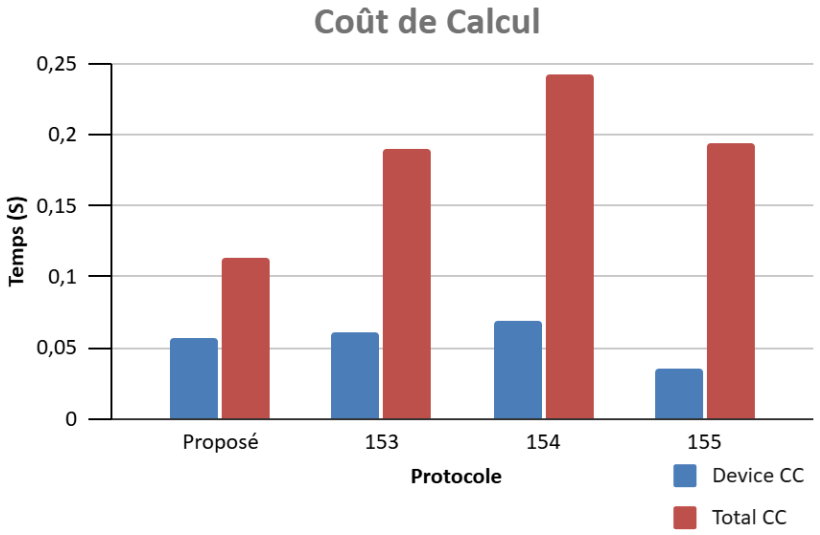


Figure. 5.10. Comparatif des coûts computationnels.

2.1.3.3. Latence moyenne

La surcharge de latence moyenne (AoD : The Average overhead delay) est une métrique importante à considérer pour évaluer l'influence de la surcharge de communication introduite par le protocole proposé pour la sécurité des communications. Dans notre évaluation, cette latence inclut la mise en file d'attente de la couche MAC ainsi que les retards de transmission et de propagation. Elle est exprimée pour un paquet comme la différence entre le moment de la réception du paquet à la destination (rec time) et le moment de l'émission du paquet du côté source (émis time).
La surcharge de latence moyenne (AoD) pour un réseau de N dispositifs est calculée comme suit :

$$AoD = \sum_{j=1}^{N} ((\sum_{i=1}^{M} rec\_time(i) - emis\_time(i))/M)/N \quad (1.3)$$

M : nombre de paquets (surcharge) introduits dans le processus d'authentification du dispositif j.

2.1.3.4. Consommation énergétique

Le modèle décrit dans [156] et connu sous le nom de modèle de communication radio de premier ordre est adopté dans notre évaluation. E_{elec} est supposé être la consommation d'énergie radio pour transmettre ou recevoir 1 bit de données, la dissipation d'énergie pour transmettre 1 bit de données du nœud source au nœud de destination à la distance d peut être estimée par l'équation suivante :

$$E_{\alpha} = l.E_{elec} + l.E_{amp}.d^{2}_{toBS} \quad (1.4)$$

La dissipation totale d'énergie par la transmission de M messages par un dispositif :

$$E_{tx\_tot} = \sum_{i=1}^{M} l_i.E_{elec} + l_i.E_{amp}.d^{2}_{j\ toBS} \quad (1.5)$$

Alors que la dissipation d'énergie par un dispositif pour recevoir l bits de données est formulée comme suit :

$$E_{rx} = l.E_{elec} \quad (1.6)$$

La dissipation totale d'énergie par un dispositif pour la réception de M' messages est :

$$E_{rx\_tot} = \sum_{i=1}^{M'} l_i.E_{elec} \quad (1.7)$$

L'énergie totale consommée par le processus d'authentification des N dispositifs dans notre cas peut être estimée par l'équation suivante :

$$E_{tot} = \sum_{j=1}^{N} (\sum_{i=1}^{M} l_i.E_{elec} + l_i.E_{amp}.d^{2}_{j\ toBS} + \sum_{i=1}^{M'} l_i.E_{elec}) \quad (1.8)$$

où :
li : la taille totale en bits des données transmises par le dispositif i.
E_{elec}: la dissipation d'énergie radio.
E_{amp} : la dissipation d'énergie de l'amplificateur de transmission.
N : le nombre de dispositifs dans le réseau.

Le modèle est adapté pour de courtes distances. Pour des distances plus longues (d > d0), le terme d^2 est remplacé par d^4 dans les équations (4), (5) et (8). d0 est la distance de croisement telle que définie dans le modèle de communication radio [156].

2.1.5. Résultats de simulation

Pour évaluer les performances du protocole proposé en termes de latence moyenne, de taux de livraison des paquets et de consommation d'énergie, nous avons réalisé des simulations

sous le simulateur NS3 sur une station de travail Linux. La station de base est située à l'origine du système de coordonnées. Les appareils intelligents sont placés à une distance de 1 à 100 m de la station de base (nous plaçons un appareil tous les deux mètres).
La communication est considérée à travers le IEEE 802.15.4 LWPAN et 2,405 GHz (canal 11). La consommation d'énergie pour l'électronique de l'émetteur et du récepteur est de 50 nJ/bit et pour l'amplificateur de l'émetteur est de 100 pJ/bit/m2. Les autres paramètres sont supposés être à leurs valeurs par défaut telles que définies dans NS3. Les paramètres de simulation sont présentés dans la Table 5.5.

Table 5.5 : Paramètres de simulation.

Paramètre	**Description**
Plateforme	NS3.33/Ubuntu 20.04
Programmation	C++
Puissance de transmission (dBm)	0 dBm
Eelec	50 nJ/bit
Eamp	100 pJ/bit/m^2
Distance maximale	100 m
Nombre d'appareils	50

Des expériences, en considérant un réseau d'appareils M2M IEEE 802.15.4 comme étude de cas, la latence moyenne, le taux de livraison des paquets (PDR) et la consommation d'énergie supplémentaire du réseau, obtenus à partir de la simulation sont présentés dans les Figure. 5.11 et 5.12.
Les résultats de simulation illustrés par le graphique dans la Figure. 5.11 montrent que la latence augmente avec l'évolution de la distance. Dans notre expérimentation, 92 m est la distance maximale couverte par la radio de l'appareil ; sur des distances plus longues (d > 92 m), le taux de livraison des paquets (PDR) devient égal à zéro. La raison en est que la puissance de transmission de 0 dB correspond à une portée de communication d'environ 87 à 92 m.

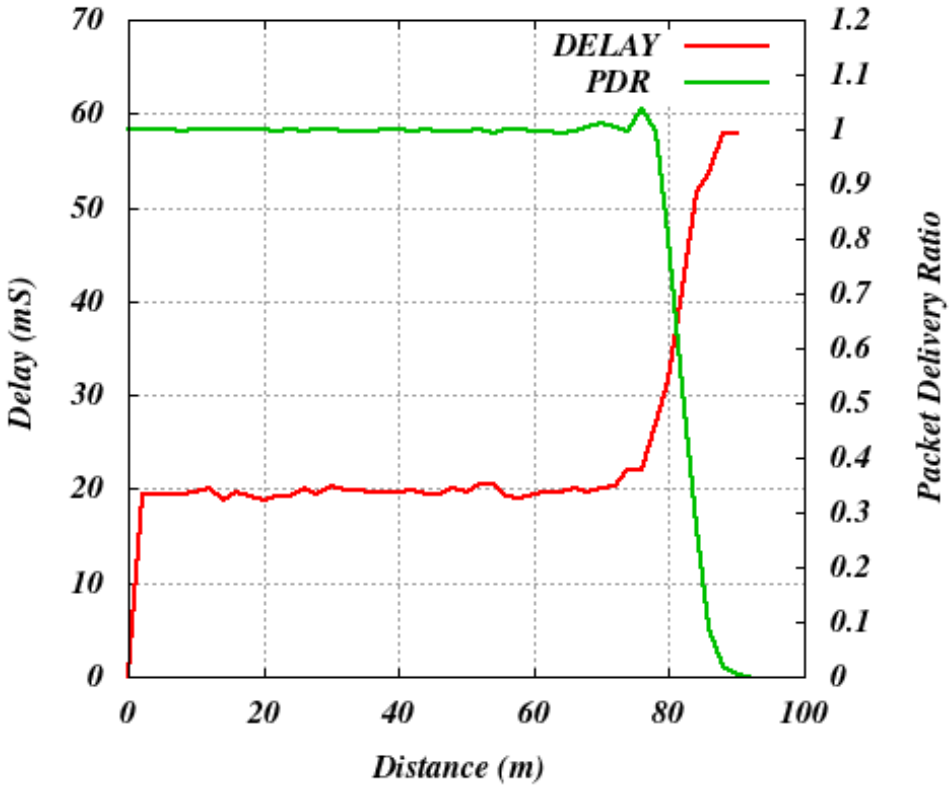


Figure. 5.11. latence moyenne, taux de livraison des paquets en fonction de la distance.

Le protocole proposé offre des surcharges de communication et de stockage comparables ainsi qu'un taux de livraison des paquets (PDR) similaire aux autres protocoles étudiés, mais avec une consommation d'énergie relativement moindre par les appareils sans fil, comme le montre la Figure. 5.12. Cela conduit par conséquent à une amélioration de la durée de vie du réseau d'appareils M2M.

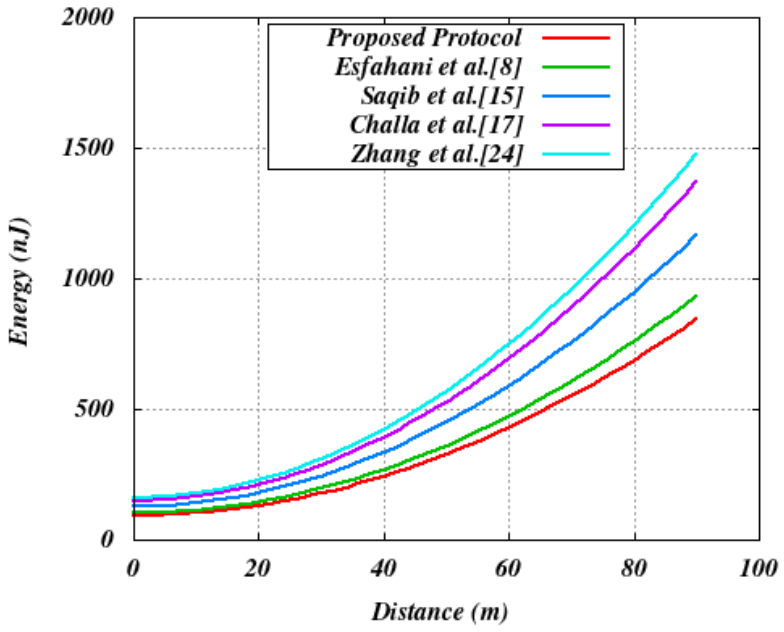


Figure. 5.12. Consommation d'énergie.

2.1.6. Analyse de sécurité

Pour vérifier la validité de notre schéma, une analyse informelle est menée. Nous utilisons un modèle d'adversaire courant [154] [156] décrivant des attaques de sécurité bien connues au niveau du réseau de périphériques M2M, et expliquons comment le schéma proposé les contrecarre. On suppose qu'un attaquant malveillant peut intercepter secrètement les communications ou les altérer entre deux entités. Il peut également usurper l'identité d'un autre périphérique afin de créer de fausses demandes et transactions, entraver la communication, rejeter les transactions, supprimer ou modifier des données de transaction, ou lier une transaction utilisateur à son identité. Un adversaire peut également transmettre un flux de messages précédemment transmis aux autres. L'adversaire peut également tenter de falsifier la signature d'une entité légitime et l'envoyer à d'autres. On suppose que les stations de base sont sécurisées.

- Attaque de l'homme du milieu : En plus de l'authentification mutuelle fournie par notre schéma, l'utilisation de fonctions de hachage par le périphérique pendant toutes les phases est considérée comme une preuve de la validité de l'identité du périphérique. Chaque fois que la station de base reçoit un message du périphérique, elle récupère ses informations à partir de la blockchain et les hache afin de comparer le résultat au code de hachage reçu. Le résultat de la comparaison montre si les données ont été modifiées ou non par un homme du milieu.
- Attaque de rejeu : l'attaquant retarde ou renvoie frauduleusement une demande au destinataire. Nous nous attendons à ce qu'un périphérique authentique ait envoyé M5 à la station de base. Dans le cas où un adversaire tente d'usurper l'identité du périphérique authentique en rejouant M5, la station de base rejettera la demande d'inscription car le périphérique est déjà enregistré. Dans le cas où un adversaire tente d'imiter le périphérique authentique en rejouant M7, la station de base calcule la différence entre le timestamp reçu et son timestamp actuel, la station de base ignore le message reçu si la valeur est hors de portée.
- Attaque d'usurpation : Nous considérons qu'un périphérique B a l'intention d'usurper l'identité du périphérique A. Cependant, le périphérique B ne peut pas obtenir le MAC du périphérique A. Si le périphérique B veut générer lui-même le MAC du périphérique A, il aura besoin du hachage précédent du bloc où le périphérique A était stocké dans la phase de pré-enregistrement, ce qui ne serait pas possible car les périphériques n'ont pas accès à la blockchain, et il n'y a pas de fonction dans le contrat intelligent déployé, qui renvoie un MAC sauf dans la phase d'inscription.
- Authentification mutuelle : L'authentification mutuelle fait référence à deux entités qui s'authentifient mutuellement avant l'établissement formel de la communication. À la réception de M7, la station de base récupère les informations du périphérique à partir de la blockchain et les hache afin de comparer le hachage à Z reçu dans M7. Le périphérique est considéré comme authentifié si la comparaison est correcte. Lorsque le périphérique reçoit le message M8, qui contient la clé publique de la BS et la clé de session, précédemment calculée par la BS, le périphérique calcule la clé de session et la compare à la clé de session reçue dans M8, la station de base est également considérée comme authentifiée si la comparaison est correcte.
- Attaque de falsification : l'adversaire tente de falsifier la signature d'une entité légitime et de l'envoyer à d'autres. Dans le schéma proposé, la clé prépartagée est échangée dans un canal

sécurisé lors de la phase de pré-enregistrement. Dans les autres phases, la PSK est toujours cryptée avant tout échange. Tant qu'elle n'est pas compromise, la théorie cryptographique adoptée garantit qu'il est impossible de falsifier une signature valide sans connaître la clé prépartagée.

2.1.7. Discussion

La Table 5.6 ci-dessous illustre les forces et les faiblesses des différents protocoles de sécurité tout en mettant en évidence les propriétés du schéma proposé basé sur la blockchain.
En plus de meilleures performances au niveau du périphérique (c'est-à-dire, faible coût de calcul, faibles surcharges de communication et de stockage utilisant un nombre relativement limité de messages courts échangés), la proposition utilise des clés cryptographiques plus longues de 160 bits, comparativement à d'autres protocoles utilisant des valeurs de hachage de 128 bits pour les fonctions cryptographiques[91]. Cela conduit à une solution cryptographique plus robuste (les clés publiques ECC de 160 bits sont équivalentes au système cryptographique RSA de 1024 bits [154]). Comparé à notre schéma, le point faible du schéma présenté dans [8] est sa consommation d'énergie plus élevée, comme le montre la Figure. 5.12.
Le schéma proposé bénéficie des avantages offerts par la technologie de la blockchain, ce qui permet un processus d'authentification plus puissant et un partage et un stockage de données plus sécurisés tout en préservant leur intégrité, leur disponibilité et leur traçabilité (ou historique des transactions).
La plupart des protocoles d'authentification proposés pour sécuriser les périphériques sans fil contraints garantissent un accord sur les clés et une authentification mutuelle, mais présentent l'inconvénient de SPF et une architecture centralisée (organisée autour d'un AS: serveur d'authentification comme dans [91], RA: Autorité d'enregistrement dans [81], GN: nœud de passerelle dans [154], ou courtier dans [155]). Au contraire, plutôt que d'utiliser l'autorité d'une organisation centralisée, le schéma proposé, en tirant parti des technologies de la blockchain et comparable au système proposé dans [8], est décentralisé et exempt de SPF. Les fonctionnalités de sécurité de la blockchain garantissent manifestement un haut niveau de confiance sans nécessité de gérer la confiance entre les périphériques et les stations de base.
Évidemment, le schéma basé sur la blockchain consomme plus de temps de traitement, lié aux opérations supplémentaires de cryptominage et de hachage, ce qui augmente donc la consommation d'énergie au niveau des stations de base. Ces dernières sont considérées comme des périphériques à ressources élevées, responsables de l'extraction et du stockage du grand livre distribué en tant que blockchain privée et sécurisée avec des coûts de calcul et de communication supplémentaires relativement faibles au niveau du réseau de périphériques.

Table 5.6 : Comparaison avec d'autres protocoles

Protocole	**Proposé**	**[91]**	**[153]**	**[81]**	**[154]**	**[155]**	**[8]**
Centralisé	Non	Oui	Oui	Oui	Oui	Oui	Non
Niveau de confiance	Élevé	Faible	Faible	Moyen	Faible	Faible	Élevé
Point unique de défaillance	Non	Oui	Oui	Oui	Oui	Oui	Non
Authentification mutuelle	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Accord de clé	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Résistance à l'homme du milieu	Oui	Non	Oui	Oui	Oui	Non	Oui
Historique des transactions	Oui	Non	Non	Oui	Non	Non	Oui

2.1.8. Conclusion

Dans le cadre de nos recherches, nous avons proposé une architecture sécurisée pour les communications de la couche M2M. Un mécanisme de sécurité est suggéré pour l'authentification mutuelle entre les stations de base et les dispositifs finaux une fois enregistrés. Une technologie de blockchain est utilisée pour fournir la sécurité des communications M2M tout en améliorant la gestion des identités et en facilitant la traçabilité, l'intégrité et la disponibilité des transactions de données.
La proposition repose sur des procédures légères impliquant un stockage minimal encombrant et un nombre limité de messages échangés pour la gestion des identités. De plus, l'analyse de sécurité de notre proposition confirme que notre solution résiste à différentes attaques. Dans le cadre des travaux futurs, nous prévoyons d'étendre le schéma proposé afin de prendre en charge la mobilité des appareils et l'hétérogénéité des réseaux. De plus, nous prévoyons d'évaluer les performances du schéma basé sur la blockchain étendu à travers des simulations, dans lesquelles les appareils doivent accéder au service d'authentification à travers les niveaux élevés de l'architecture M2M, et de mettre en œuvre un nouveau consensus léger PBFT adapté.

2.2 Mécanisme d'authentification léger basé sur une blockchain à deux couches pour les communications M2M dans l'IoT (*Blockchain-based Authentication Protocol for Wireless M2M area Networks with Sidechain Integration*)

Cette contribution introduit une approche à double blockchain, comprenant une Mainchain et une Sidechain, pour répondre aux problèmes d'évolutivité (scalabilité ou mise à l'échelle) qui prévalent dans les solutions blockchain traditionnelles.

2.2.1. Introduction

Étant donné que les dispositifs contraints en ressources sont mal adaptés pour gérer les demandes computationnelles de la technologie de la blockchain, une approche alternative est adoptée qui propose un protocole d'authentification basé sur un travail précédent [157]. Le protocole exploite le potentiel d'une blockchain pour stocker les nœuds initialement enregistrés, principalement à des fins d'authentification, avant d'initier l'échange de données pendant la phase de communication. Dans cette approche, deux blockchains distinctes (c'est-à-dire une chaîne principale et une chaîne latérale) sont mises en œuvre et positionnées stratégiquement au niveau d'un ensemble de stations de base situées au niveau du Edge, près du champ de détection, à savoir la zone M2M conformément aux normes architecturales de l'ETSI [19], comme illustré dans la Figure. 5.13.

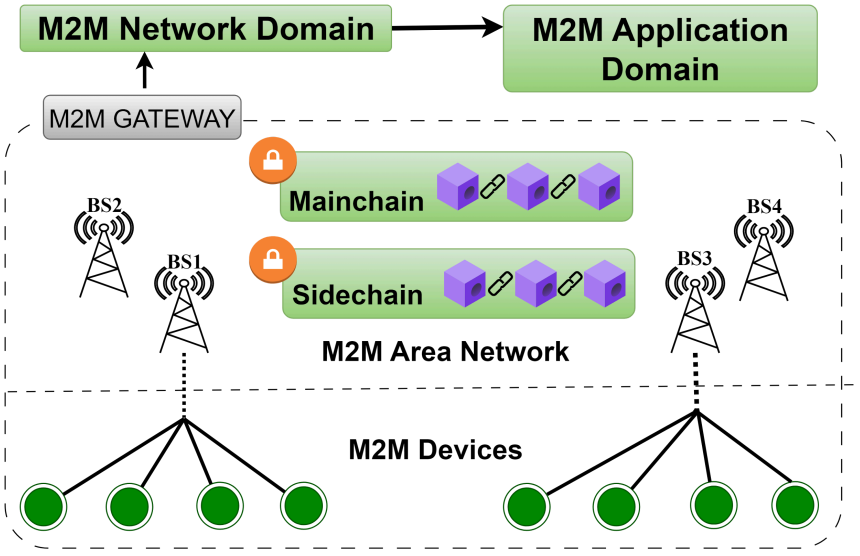


Figure. 5.13. Architecture M2M basée sur la blockchain.

Cette approche présente plusieurs avantages, notamment la réduction des retards de traitement et l'énergie consommée par les nœuds du réseau. Le cadre de la blockchain est conçu pour stocker toutes les informations liées aux appareils individuels.

Nous avons désigné à la fois la chaîne principale et la chaîne secondaire comme privées, ne stockant que des informations limitées sur les dispositifs. Contrairement aux blockchains publiques qui privilégient la transparence et l'immutabilité, notre approche met l'accent sur la confidentialité des données et l'accès contrôlé. La Figure. 5.14 illustre les structures de blocs des deux blockchains.

En conjonction avec le système de communication sécurisé, nous proposons un pont robuste, reliant notre chaîne principale à une chaîne secondaire dédiée. Ce pont agit comme un conduit pour le flux transparent de données et d'actifs entre les deux chaînes, leur permettant de fonctionner en concert tout en préservant les attributs distincts de chacune, comme le montre visuellement la Figure. 5.14.

Le pont comprend des contrats intelligents qui exécutent des fonctions prédéfinies et limitées, facilitant la synchronisation et l'échange de données et d'actifs entre la chaîne principale et la chaîne secondaire. Ce protocole de pont proposé est hautement adaptable, capable de s'intégrer de manière transparente à un large éventail de blockchains, qu'elles soient publiques ou privées et utilisent différents mécanismes de consensus. Cette adaptabilité est un choix de conception délibéré pour garantir la compatibilité avec des configurations de blockchain diverses.

La clé de cette adaptabilité réside dans l'utilisation du langage de programmation approprié pour la blockchain choisie. Les contrats intelligents qui forment le pont peuvent être écrits dans différents langages de programmation spécifiques à ces blockchains, car ils peuvent être hébergés indépendamment sur différentes plateformes. Dans notre cas, où nous utilisons Ethereum [106] pour héberger les deux blockchains, nous avons choisi Solidity, le langage de programmation couramment utilisé pour les contrats intelligents sur Ethereum.

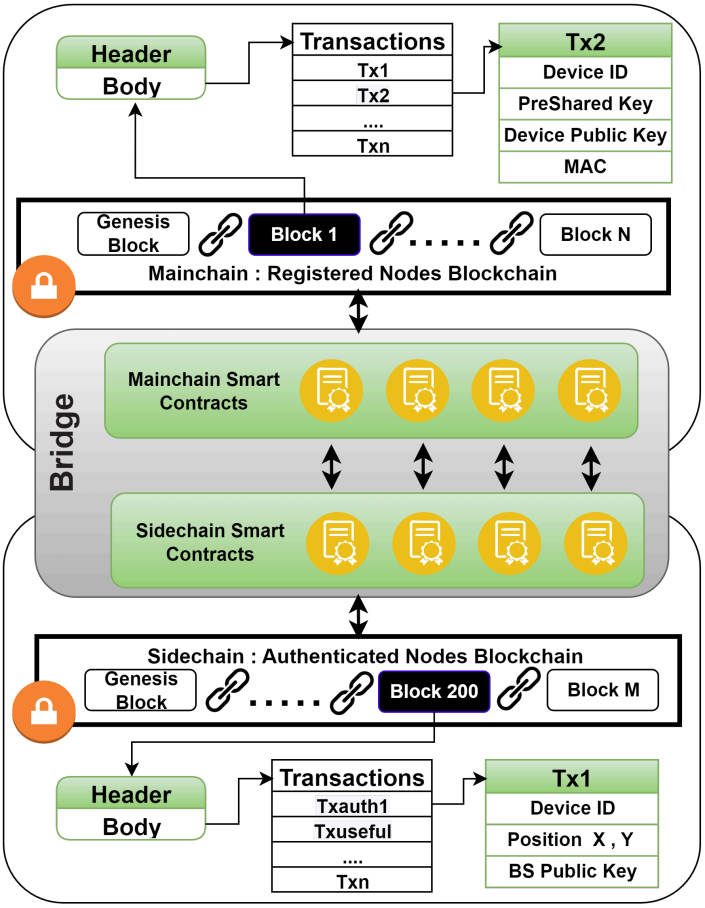


Figure. 5.14. Structure des blockchains

Pour garantir une communication sécurisée, le dispositif et la station de base (BS) doivent établir une clé de session. À cette fin, le protocole d'échange de clés Elliptic Curve Diffie-Hellman (ECDH) [74] est utilisé. Ce protocole dérive la clé secrète partagée (SK) à partir du point de base de la Cryptographie à Courbes Elliptiques (ECC) utilisée.

2.2.2. Processus d'authentification

Le processus d'authentification au sein de ce protocole basé sur la blockchain est structuré en trois phases clés : Enregistrement, Autorisation et Authentification.

2.2.2.1. Phase d'enregistrement

Dans la phase d'enregistrement, chaque dispositif débute en générant sa clé privée (DPrK), choisie aléatoirement dans la plage [1, n - 1], où 'n' représente l'ordre de la courbe et 'G' désigne le point de générateur sur la courbe elliptique. Ensuite, le dispositif calcule sa clé publique (DpbK) en utilisant (2.1).

$$DPbK = DPrK * G \qquad (2.1)$$

Parallèlement à cela, une clé pré-partagée (PsK) est créée pour l'enregistrement du dispositif. À la fois la PsK et la paire de clés du dispositif (DPrk, DPbk) sont ensuite chargées dans le dispositif lors de sa configuration initiale.
Ensuite, la station de base ajoute les informations du dispositif, y compris sa clé publique (DPbk), l'identifiant du dispositif, et la PsK chiffrée en utilisant le Dpbk, à la blockchain privée (chaîne principale). Ce processus est effectué en appelant la fonction registerDevice() du contrat intelligent déployé sur la chaîne principale.

2.2.2.2. Phase d'autorisation

Le processus d'autorisation interagit exclusivement avec la chaîne principale et se compose de neuf étapes séquentielles, comme illustré dans la Figure.5.15.
Les étapes détaillées sont les suivantes :
1- Le dispositif calcule la signature en chiffrant l'identifiant du dispositif en utilisant sa propre clé privée.

2- Le dispositif envoie l'identifiant du dispositif (DID) et la signature à la station de base.

3- La station de base lance un processus de vérification en interagissant avec la chaîne principale. Elle vérifie la présence du dispositif sur la chaîne principale en utilisant la fonction IsRegistered() du contrat intelligent de la chaîne principale. Si le dispositif n'est pas trouvé, l'autorisation est refusée. Sinon, la fonction GetInformation() est appelée pour récupérer les informations du dispositif, y compris la clé publique du dispositif et la clé pré-partagée (PsK) chiffrée.

4- La station de base déchiffre la signature en utilisant la clé publique du dispositif, puis compare le résultat avec l'identifiant du dispositif. Ce processus garantit l'intégrité et l'authenticité du message. Si la comparaison échoue, indiquant que le message a été altéré ou

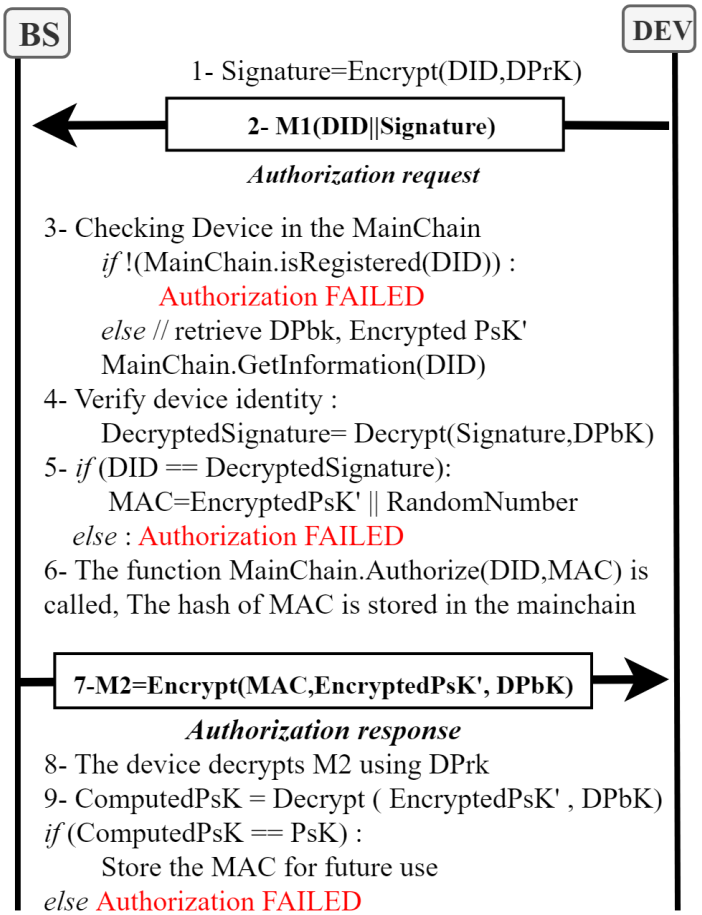


Figure. 5.15. Diagramme du processus d'autorisation

qu'il provient d'une source non autorisée, le processus d'autorisation est terminé ; sinon, il passe à l'étape suivante.

5- Le Code d'Authentification de Message (MAC) est dérivé de la PsK chiffrée stockée dans la blockchain concaténée avec un nombre généré aléatoirement.

6- La fonction AuthorizeDevice (DID, MAC) de la chaîne principale est exécutée, ce qui stocke le hachage du MAC du dispositif dans la chaîne principale pour référence future.

7- La station de base chiffre le MAC, la PsK chiffrée récupérée de la blockchain en utilisant la clé publique du dispositif, puis envoie ce message chiffré M2 au dispositif.

8- Le dispositif déchiffre le message en utilisant sa propre clé privée pour récupérer le MAC et la PsK' chiffrée.

9- Le dispositif déchiffre la PsK chiffrée en utilisant sa clé privée, puis compare le résultat avec la PsK' stockée dans le dispositif. Si la comparaison est correcte, l'autorisation est accordée; sinon, la station de base est compromise et l'autorisation échoue.

2.2.2.3. Phase d'authentification

Le schéma détaillé du protocole décrivant la phase d'authentification est présenté dans la Figure. 5.16.

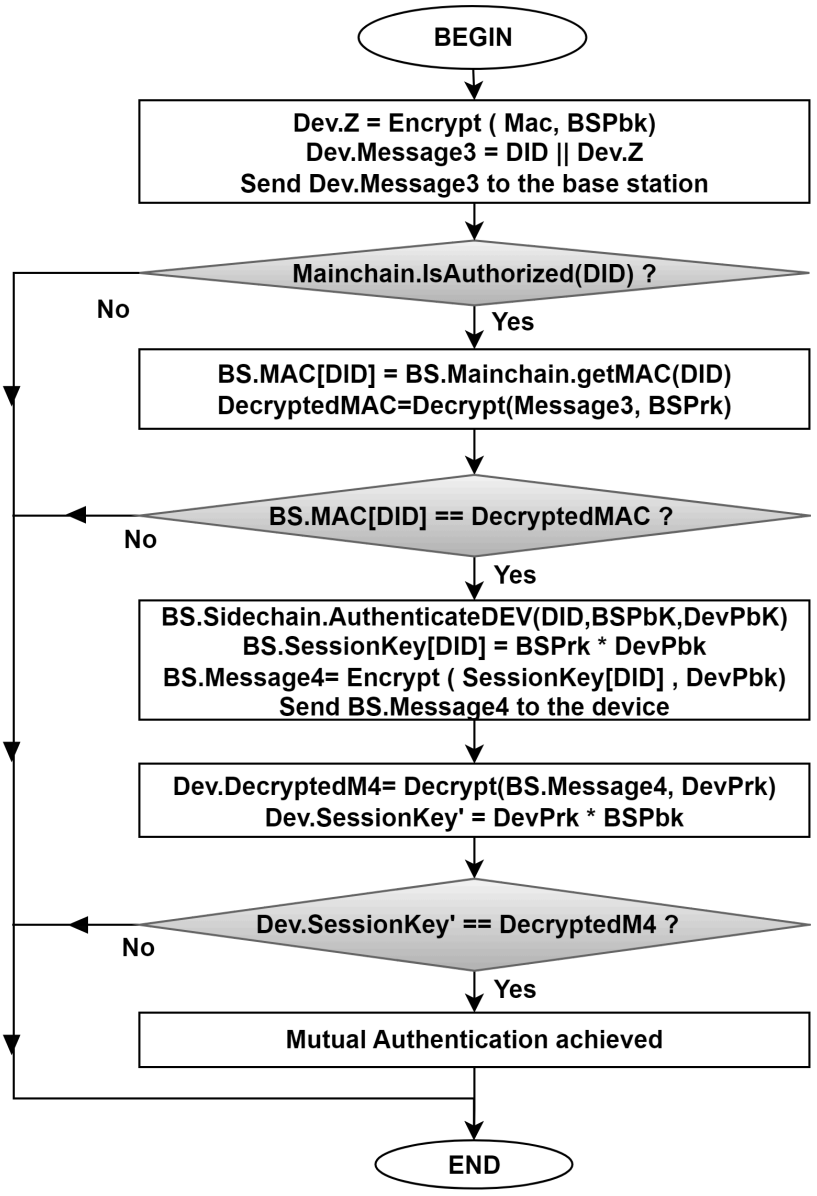


Figure. 5.16. Organigramme du processus d'authentification

Le processus d'authentification commence par le dispositif chiffrant le message MAC en utilisant la clé publique de la station de base (BSPbK) et stocke le résultat dans la variable Dev.Z. Ensuite, le dispositif construit le message M3 en concaténant son DID avec Dev.Z, et transmet M3 à la BS.

À la réception de M3, la station de base initie le processus d'authentification en vérifiant le statut d'autorisation du dispositif sur la chaîne principale. Si le dispositif est autorisé, la station de base récupère le MAC associé au dispositif depuis la chaîne principale en utilisant son DID. Ensuite, elle déchiffre Dev.M3 en utilisant sa clé privée (BSPrk) pour obtenir le MAC déchiffré.

Si le MAC stocké dans la chaîne principale correspond au MAC déchiffré, indiquant un déchiffrement et une vérification réussis, l'authentification réussit. Sinon, l'authentification échoue et le processus est terminé. Ensuite, la station de base authentifie le dispositif sur la sidechain en invoquant la fonction AuthenticateDev, fournissant l'ID du dispositif, sa propre clé publique (BSPbK) et la clé publique du dispositif (DevPbK).
Après une authentification réussie, la station de base calcule la clé secrète partagée (SK). En utilisant (1) et étant donné que la station de base dispose déjà d'une clé privée et d'une clé publique, elle dérive SK comme indiqué dans (2.2) . La station de base chiffre ensuite SK pour générer le message M4, qui est envoyé au dispositif.

$$SK = BSPrK * DPbK = BSPrK * (DPrK * G) = DPrK * (BSPrK * G) = DPrK * BSPrK \quad (2.2)$$

À la réception de M4, le dispositif le déchiffre en utilisant sa clé privée (DPrk). Le dispositif calcule ensuite sa propre clé secrète partagée (Dev.SK') selon (2) en utilisant sa clé privée (DPrK) et la clé publique de la station de base (BSPbK). Si Dev.SK correspond à SK reçu de la station de base, l'authentification est réussie. Sinon, l'authentification échoue.

2.2.3. Évaluation des performances

Dans notre étude, nous avons comparé les performances de notre protocole avec celles proposées dans notre précédent travail [157]. Cette comparaison peut être étendue à des protocoles basés sur une blockchain à une seule couche [8][158][159]. Tous ces travaux présentent des problèmes de scalabilité. Dans [157], une seule chaîne est utilisée et sera désignée dans le reste du document comme BC1. Les deux chaînes utilisées dans ce travail seront désignées par MC2 pour la chaîne principale et SC2 pour sidechain.

Dans [158], une blockchain privée à l'échelle industrielle pour les appareils IoT est proposée pour préserver la confidentialité des propriétaires d'appareils tout en préservant les identités des utilisateurs via la blockchain et le cloud computing. La plateforme proposée est basée sur le protocole EPID (Enhanced Privacy Identity). Une architecture ChainAnchor est introduite pour soutenir le commissionnement anonyme des appareils. Ce système permet aux propriétaires d'appareils de recevoir une compensation pour la vente de données collectées aux fournisseurs de services, tout en permettant également aux fournisseurs de services de partager des données de capteurs de manière préservant la confidentialité.

Pour protéger la confidentialité et les exigences de sécurité des appareils IoT dans les maisons intelligentes, les auteurs dans [159] suggèrent une approche basée sur une blockchain privée. L'architecture BC proposée intègre “Hyperledger Fabric” et “Hyperledger Composer” pour surmonter les limitations de sécurité de la plupart des approches basées sur la blockchain privée et répondre aux exigences de sécurité des maisons intelligentes.

2.2.3.1. Coût de stockage:

Dans [157], les transactions d'inscription (TXreg), les transactions d'autorisation (TXatrz), les transactions d'authentification (TXauth), et plusieurs autres transactions utiles (TXuseful) peuvent être stockées dans le même bloc de la blockchain (BC1). Nous supposons que chaque nœud a effectué au moins une inscription, une autorisation et une authentification. Cela entraîne à son tour une augmentation de la taille de la blockchain BC1, qui dépassera la taille de la chaîne principale (MC2) proposée dans ce protocole. Cela peut être démontré à travers les équations suivantes :

$$\text{Len(BC1)}=\sum_{0}^{N}(TXreg + TXatrz + TXauth + TXuseful) \qquad (2.3)$$

En revanche, notre nouveau protocole introduit le concept d'une sidehcain. (5) et (6) définissent respectivement la longueur de la sidechain (SC2) et la chaîne principale (MC2) comme suit :

$$\text{Len(SC2)} = \sum_{0}^{N}(TXatrz + TXauth) \qquad (2.4)$$

$$\text{Len(MC2)} = \sum_{0}^{M}(TXreg + TXuseful) \qquad (2.5)$$

en combinant (2.3), (2.4) et (2.5), nous obtenons :

$$\text{Len(BC1)}=\text{Len(MC2)}+\sum_{0}^{N}(TXatrz + TXauth) \qquad (2.6)$$

Puisque les nœuds sont authentifiés et autorisés au moins une fois :

$$\sum_{0}^{N}(TXatrz + TXauth) > 0 \qquad (2.7)$$

À partir de (2.6) et (2.7), nous déduisons que Len (BC1) > Len(MC2)

2.2.3.2. Analyse du Temps d'Authentification

Nous simulons une demande d'autorisation et une demande d'authentification. Le temps d'authentification peut être divisé en plusieurs composantes. Des facteurs tels que la congestion du réseau et la puissance de calcul des nœuds peuvent potentiellement affecter le temps d'écriture. Cependant, dans le cadre de notre simulation et de notre comparaison, nous négligeons ce paramètre pour garantir une évaluation cohérente des blockchains sélectionnées.

1)Temps de calcul : Cela inclut le temps nécessaire pour divers calculs.

2)Temps de communication : Le temps nécessaire aux processus de communication.

3)Temps d'écriture des données sur la blockchain : Il est important de reconnaître que le temps nécessaire pour écrire des transactions sur une blockchain peut varier en fonction des réseaux blockchain choisis. Par exemple, le temps estimé pour écrire des transactions sur la blockchain Ethereum est d'environ 15 secondes [106].

4) *Temps de lecture des données de la blockchain :* Le temps de lecture peut être calculé comme suit :

$$Tresp(N, B) = (N * B) / S \qquad (2.8)$$

Ici, Tresp (N, B) représente le temps de réponse en fonction de la taille de la blockchain en termes du nombre de blocs (N) et de la taille du bloc (B). S est fixé à 10000 Tx/s pour représenter la vitesse de récupération des données de la blockchain.

2.2.4. Résultats de simulation

Les simulations ont été réalisées dans un environnement contrôlé, en utilisant un ASUS VivoBook x512 pour le matériel et le framework NS-3 sous le système d'exploitation Ubuntu 20.04. Les opérations de chiffrement symétrique sont basées sur AES-128 bits (0,004s). Les opérations de chiffrement asymétrique sont effectuées en utilisant ECDH (0,0171s). SHA-3 avec 256 bits est utilisé pour les opérations de hachage (0,00032s), et un temps de communication estimé à 0,02s [157].
Nous avons mené des simulations à la fois sur BC1 et MC2, en considérant des tailles de blockchain de 1000 blocs et 2000 blocs. Lorsqu'un nœud lance un processus d'authentification, la station de base doit vérifier sa présence dans la blockchain, BC1, comme décrit dans [157], ou la chaîne principale, MC2, dans notre cas particulier. Nous avons examiné trois positions de bloc différentes pour chaque scénario de longueur de bloc : lorsque TXreg est situé dans le premier bloc, le bloc du milieu et le dernier bloc de BC1/MC2, désignés respectivement par FirstBlock, MidBlock et LastBlock.
La Figure. 5.17 illustre une comparaison des durées d'authentification à travers les trois positions de bloc mentionnées précédemment, en tenant compte des différentes longueurs de blockchain. Le graphique montre que les blockchains à deux couches (MC2) présentent des durées d'authentification plus courtes par rapport aux blockchains à une seule couche (BC1).

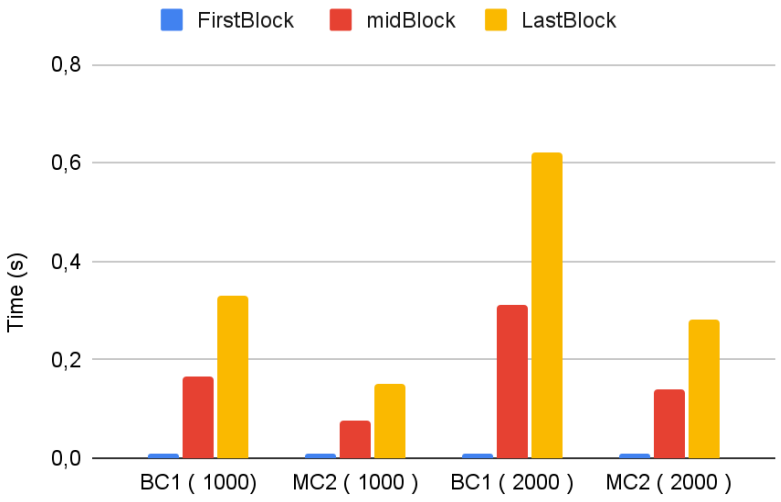


Figure. 5.17. Comparaison du temps d'authentification.

Le graphique de la Figure. 5.18 illustre clairement que l'augmentation du nombre de nœuds n'affecte pas significativement le temps de transaction d'authentification, qui reste plus faible dans le cas du protocole basé sur la sidechain par rapport au modèle de blockchain traditionnel [8][154][155]. Cela est dû à ses avantages dans la réduction de la complexité de la récupération des informations.

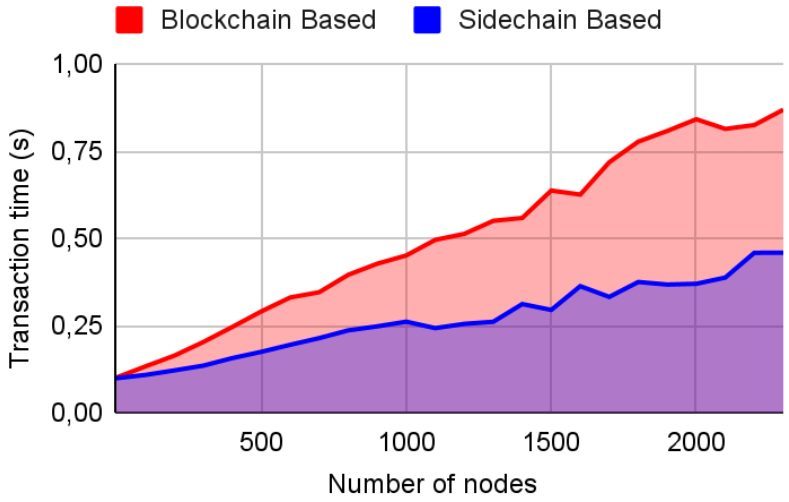


Figure. 5.18. Temps d'authentification vs nombre de nœuds.

Les résultats illustrés dans la Figure. 5.17 et la Figure. 5.18 montrent l'efficacité de notre approche de sidechain dans la gestion de plus grandes quantités de données, et démontrent l'efficacité des sidechains dans la résolution des problèmes de scalabilité. Cela est réalisé en soulageant la congestion sur la mainchain grâce au stockage des transactions non essentielles dans la sidechain.

2.2.5. Analyse de sécurité

Dans cette étude, AVISPA [10] est utilisé pour évaluer la résistance de notre protocole aux menaces actives, notamment les attaques de rejeu et de l'homme du milieu, confirmant l'authentification mutuelle et la confidentialité des clés privées. La spécification de propriétés de sécurité basée sur HLPSL, comme représenté dans la Figure. 5.19, englobe l'assurance de la confidentialité pour le Mac (secret de Sec_1), la clé de session Sk (secret de Sec_2), et l'authentification mutuelle entre le dispositif et la station de base (authentification_sur sk_auth_1).

```
goal
        secrecy_of secrecy_MAC,secrecy_SK
        %%for Mac, SessionKey secrecy
        authentication_on sk_auth_1
        %% Mutual Authentication
end goal
```

Figure. 5.19. Objectifs et propriétés de sécurité

A. Attaque de rejeu : OFMC évalue si une partie légitime pourrait agir involontairement en tant qu'intrus passif en exécutant un schéma spécifique. Par la suite, OFMC fournit à cet intrus des informations sur les sessions régulières menées par des agents légitimes. De plus, OFMC évalue la possibilité pour l'intrus de mener une attaque de l'homme du milieu.
Dans la Figure. 5.20, deux sessions identiques ont été intentionnellement configurées pour simuler l'occurrence d'une attaque de rejeu. La spécification fournie ci-dessous pour le rôle de l'environnement dans HLPSL décrit comment ces deux sessions identiques entre le dispositif et la station de base sont traitées, permettant la détection des attaques de rejeu si elles se produisent.

```
intruder_knowledge = {alice,bob,ka,kb}
composition
session(alice,bob,s1,ka,kb,psk)
session(alice,bob,s1,ka,kb,psk) %% For Replay Attack
session(i,bob,s1,ka,kb,psk) %% For the intruder MIM
```

Figure. 5.20. Les principales spécifications.

Les résultats présentés dans la Figure. 5.21, démontrent que notre protocole est sûr, comme indiqué par le résultat.

B. Attaque de l'homme du milieu : Le troisième scénario commenté dans le rôle de l'environnement dans la Figure. 5.20, est conçu pour détecter la présence d'une attaque de l'homme du milieu, si elle se produit. Dans notre approche, le protocole ECDH (Elliptic Curve Diffie-Hellman)[74] est employé pour le partage de clés afin d'établir une connexion sécurisée. De plus, le dispositif utilise sa clé privée pour signer son message de demande d'autorisation pour qu'il puisse s'authentifier dans le réseau, ce qui assure l'intégrité des messages échangés. Les résultats, représentés dans la Figure 5.21, fournissent des preuves que notre protocole est sécurisé contre les attaques de l'homme du milieu et assure une authentification mutuelle.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/Bilami.if
GOAL
  as_specified
BACKEND
  OFMC
```

Figure. 5.21 Résultats de la vérification du protocole

2.2.6. Conclusion

L'évaluation du processus d'authentification proposé montre de bonnes performances en termes de temps et de consommation énergétique. L'ajout d'une sidechain permet de répondre aux défis de scalabilité inhérents aux blockchains classiques tout en optimisant l'utilisation des ressources. L'évaluation de cette approche montre que notre système peut gérer efficacement la sécurité des communications M2M en minimisant les risques et en préservant les performances de la mainchain pour les transactions essentielles. Cependant, notre conception présente certaines limitations car elle considère des réseaux d'aires de dispositifs M2M homogènes avec une faible mobilité, ce qui limite son application aux environnements hétérogènes dynamiques avec une forte mobilité.

En tant que travaux futurs, nous prévoyons d'étendre cette proposition à la conception complète d'un protocole combiné léger basé sur la blockchain tout en garantissant plus de services pour sécuriser les communications M2M dans des environnements hétérogènes dynamiques.

2.3. Protocole de communication multi-sauts sécurisé et économe en énergie (*SEEM-D2D: Secure and Energy Efficient Multi-hop D2D Communications in Wireless M2M Area Networks using Two-Layer Blockchain*)

2.3.1. Introduction

Les communications multi-sauts sont reconnues pour leur efficacité énergétique par rapport aux communications directes ou à saut unique. Toutefois, dans les réseaux de communication multi-sauts, les nœuds de relais intermédiaires introduisent un nouveau degré de vulnérabilité aux menaces de sécurité. Pour répondre à ce défi, nous proposons un protocole de communication multi-sauts sécurisé et économe en énergie pour les réseaux des dispositifs sans fil M2M. Les communications peuvent fonctionner dans ce cas l'un des deux modes suivants: le mode supervisé (ou avec infrastructure) sous le contrôle des stations de base, ou le mode non supervisé (ou en mode ad hoc sans infrastructure) où les appareils communiquent directement sans supervision des stations de base (SB). Les communications non supervisées sont également connues sous le nom de communications D2D.

Les communications D2D représentent un nouveau paradigme dans les réseaux cellulaires. Elles ont récemment attiré une grande attention en raison de leur grande efficacité en termes de consommation d'énergie et d'utilisation du spectre. Les communications D2D sont devenues ainsi l'une des technologies clés des réseaux hétérogènes 5G et 6G (HetNets).

Les communications D2D peuvent être utilisées aussi pour les communications M2M afin d'améliorer les performances du réseau et de réduire les délais de transmission [2]. En effet, elles offrent de faibles retards, vu qu'elles permettent d'éviter le passage par le domaine des réseaux de communication via les stations de bases alors que les nœuds sont à proximité, pouvant communiquer directement entre eux, en utilisant un chemin de traversée du signal plus court. Diverses technologies sans fil à courte portée comme Bluetooth, Zigbee, WiFi Direct et LTE Direct, peuvent être utilisées pour assurer la communication D2D.

Les communications D2D peuvent être utilisées comme alternative dans les réseaux cellulaires dans différentes applications, par exemple dans les applications où la station de base n'est pas nécessaire, ou en cas de défaillance de la station de base comme dans une situation de catastrophe naturelle telle que le séisme. Dans ce cas, les données peuvent être acheminées d'une source à une destination à travers une liaison directe, en un seul saut ou en multi-sauts, dépendamment de la route choisie.

Le routage sécurisé est l'une des solutions adoptées pour contrer les problèmes de sécurité dans les réseaux de communication D2D multi-sauts.
Dans ce contexte, un modèle de communication multi-sauts sécurisé et économe en énergie est proposé pour réaliser les tâches suivantes:
- Échange de données entre les dispositifs M2M dans l'un des deux modes suivants :
1)- avec infrastructure sous le contrôle des stations de base
2)- sans infrastructure, en utilisant les communications D2D en cas de défaillance de la station de base.

- Sécurité des communications grâce à l'introduction d'une blockchain à deux niveaux (une chaîne principale et une chaîne secondaire) pour stocker les nœuds enregistrés au niveau de la périphérie.

La blockchain est utilisée pour garantir la sécurité en termes d'authentification, de confidentialité et de disponibilité, confidentialité et de disponibilité. En outre, la sidechain ajoutée est utilisée pour permettre l'évolutivité des réseaux de périphériques M2M.
De manière concise, le système proposé tend à assurer la sécurité et l'évolutivité des réseaux M2M, à l'aide d'une blockchain à deux couches comme proposé dans [160], et d'un protocole de routage D2D multi-sauts économe en énergie.

2.3.2. Modèle du système

Notre schéma proposé est destiné aux réseaux d'aires sans fil M2M avec une faible mobilité. Selon la norme 3GPP, les communications M2M se caractérisent par 14 fonctionnalités, notamment une faible consommation d'énergie, un faible débit, une tolérance temporelle, la transmission de petites quantités de données, une faible mobilité (c'est-à-dire que les dispositifs M2M ne se déplacent pas, se déplacent rarement, ou se déplacent uniquement dans une certaine région) [36].
En ce qui concerne l'architecture M2M générale telle que définie par l'ETSI [19], le modèle proposé est présenté dans la Figure. 5.22 et divisé en trois domaines interconnectés. Le premier est le domaine M2M (dispositifs, passerelles), suivi du domaine du réseau M2M et enfin, du domaine d'application M2M.
Les dispositifs M2M, organisés en clusters, sont utilisés pour l'acquisition et la transmission des données demandées. Un réseau de dispositifs M2M établit des connexions entre ces dispositifs et les passerelles M2M. Ces dernières, à leur tour, interconnectent les réseaux de dispositifs M2M avec des réseaux de communication plus larges. Le réseau de communication M2M sert de lien entre les passerelles M2M et Internet ou les serveurs d'application M2M.

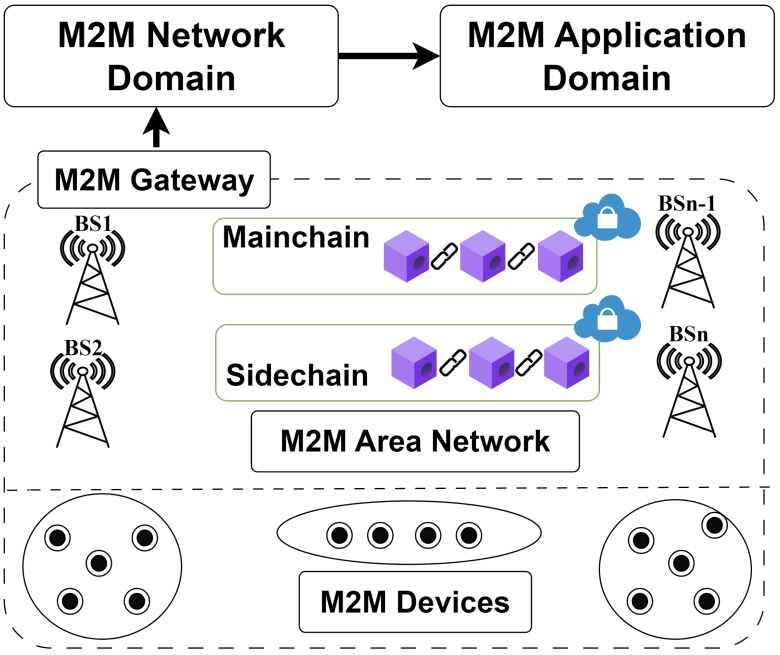


Figure. 5.22. Architecture M2M

2.3.2.1. Formation de clusters

L'adoption d'architectures en clusters ou hiérarchiques pour les réseaux sans fil, sous le contrôle des stations de base, contribue à réduire la consommation d'énergie et donc à augmenter la durée de vie du réseau. Cela se fait en éliminant les transmissions D2I redondantes, et en fusionnant également les données collectées dans un seul paquet avant de les transmettre au puits par une tête de cluster.

Cependant, la création de clusters est une opération consommatrice d'énergie. Par conséquent, le clustering centralisé est adopté dans notre schéma proposé (c'est-à-dire que l'opération est déléguée aux stations de base que nous supposons dotées d'une réserve infinie d'énergie). De plus, ce choix est renforcé pour la raison que la BS est plus sécurisée que les dispositifs autonomes et sujets à différentes attaques.

Comme dans la plupart des versions du protocole LEACH [161], nous utilisons un clustering dynamique basé sur deux phases, la phase de configuration (ou d'initialisation) et la phase stable (ou de communication).

À chaque phase d'initialisation (ou de configuration), chaque nœud du réseau, en communication montante, envoie un paquet de contrôle à sa BS, contenant son identifiant, sa réserve d'énergie et sa localisation sur le réseau. Ces informations sont utilisées par la station de base pour la définition des clusters et la sélection des têtes de cluster (CH). On suppose que les têtes de cluster sont des dispositifs hétérogènes équipés de deux antennes, permettant la communication sur deux canaux séparés. Les CH sont sélectionnées de manière simple, à chaque cycle, la BS sélectionne entre les CH potentiels de chaque cluster, celui avec la plus haute énergie résiduelle comme nouveau CH. Suite à la sélection des nouvelles têtes de cluster, les stations de base diffusent la liste mise à jour des CH avec leurs positions respectives à tous les nœuds du réseau. Ensuite, chaque dispositif sélectionne de manière autonome la CH la plus proche à laquelle il doit être associé.

À chaque cycle, le schéma TDMA (Time Division Multiple Access) est utilisé pour allouer un ensemble de créneaux horaires (time slots) appartenant à un canal commun afin d'éviter les interférences et les collisions entre les nœuds pendant les transmissions. Dans le contexte de la gestion de clusters, l'utilisation de différentes fréquences pour les clusters voisins aide à éviter les interférences et les collisions entre les transmissions de données. Chaque cluster se voit attribuer une fréquence spécifique dans le spectre radio.
De plus, le TDMA permet la préservation de l'énergie en permettant aux nœuds de désactiver leurs antennes après la transmission et d'entrer en mode veille, en attendant leur prochain créneau horaire pour transmettre.

2.3.2.2. Routage

Pendant la phase de communication, les données sont routées des nœuds vers la BS via les CH, ou de dispositif à dispositif directement sans utiliser la BS.
Le modèle proposé prend en compte deux modes de communication :

1) Les communications cellulaires classiques via les stations de base, lorsque la destination est accessible via l'une des stations de base connectant les réseaux de la zone M2M. De plus, dans la communication multi-sauts D2I/I2D (De dispositif à infrastructure, infrastructure à dispositif), un nœud peut non seulement communiquer avec sa propre BS, mais peut également communiquer avec d'autres stations de base en utilisant les voisins et les têtes de cluster comme nœuds relais. Chaque nœud agrège les données reçues et collectées et les envoie pendant son créneau horaire au prochain saut, qui pourrait être un autre dispositif, une CH ou une BS.
2) Directement entre eux en utilisant les communications D2D, les dispositifs M2M peuvent échanger des données directement de manière ad hoc sans intervention de la station de base.

Les protocoles de communication économes en énergie pour les réseaux de dispositifs sans fil IoT/M2M sont généralement basés sur un routage économe en énergie.
Dans notre conception, nous adoptons un protocole de routage en phase de communication inspiré de AODV et basé sur trois phases (Demande de route, Réponse de route et Transmission de données). De plus, chaque nœud dans le chemin de routage récupéré estime et choisit le meilleur chemin en termes de consommation d'énergie, avant de transmettre les données pendant son créneau horaire, en utilisant sa table de routage.
La sécurité des communications M2M est assurée au niveau du routage. Des fonctionnalités supplémentaires ont été intégrées à la fonction de routage pour garantir des transmissions de données sécurisées entre les dispositifs. Les détails spécifiques du mécanisme de routage sécurisé proposé sont expliqués plus tard dans la section suivante de la phase de communication à venir.

2.3.3. Système de sécurité proposé

La sécurité de notre système de communication repose sur un travail précédent [157], dans lequel nous avons introduit un protocole d'authentification léger utilisant une blockchain privée conçue pour contenir des informations relatives à chaque dispositif (identifiant du dispositif, clé pré-partagée,...). La blockchain est située sur un ensemble de stations de base plus proches du champ de détection (c'est-à-dire, la zone M2M). Pour garantir la sécurité en termes d'authentification au sein des réseaux de dispositifs M2M sans fil, le mécanisme proposé fonctionne à travers trois phases: la pré-enregistrement, l'enregistrement et l'authentification.
La phase de pré-enregistrement implique la génération d'une clé pré-partagée (PsK) pour l'enregistrement du dispositif, qui est ensuite préchargée dans le dispositif lors de sa configuration initiale.

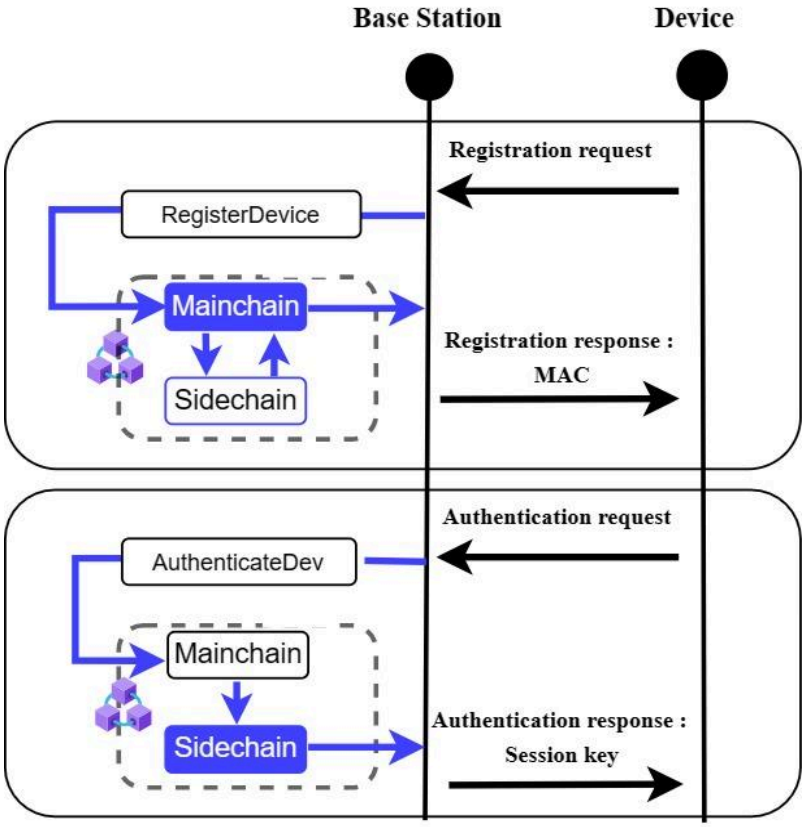


Figure. 5.23. Diagramme de séquence du protocole proposé.

Cette clé et l'identifiant du dispositif sont ajoutés à la blockchain privée lors de la phase d'enregistrement. Cet enregistrement est nécessaire pour que le nœud soit authentifié au sein du réseau, lui permettant de communiquer.
Cependant, nous avons apporté une modification significative aux étapes d'enregistrement et d'authentification. Nous avons également mis en œuvre une sidechain à la blockchain principale comme indiqué dans la Figure. 5.23, dans le but de résoudre le problème d'évolutivité de la technologie blockchain [119] et d'améliorer la sécurité des communications M2M.

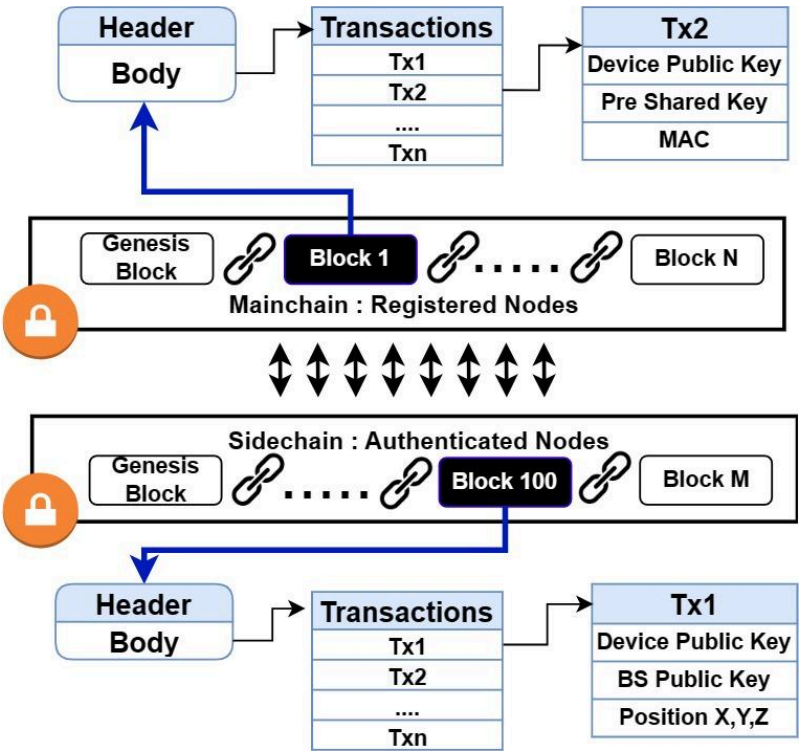


Figure. 5.24. Structure des blocs

Notre système utilise Ethereum [106] pour héberger les deux blockchains, en sélectionnant Solidity comme langage de programmation pour les contrats intelligents, largement reconnu au sein de la communauté Ethereum. La structure des deux blockchains est illustrée dans la Figure. 5.24.

2.3.3.1. Phase d'enregistrement

```
contract MainChain{
    address public SCAddress = 0x6B175474E89094C44Da98b954EedeAC4952
    SideChainContract public SideChainContractInstance;
    address Administrator; Device public device;
    uint256 public Dindex;  uint256 public allindex;
    bytes32 MAC; bytes32 PSK;
    struct Device {uint256 Dindex; uint256 DeviceID; ···
    mapping(uint => Device) public AllDevices;
    constructor () public { allindex = 0; Dindex = 0;···
    event DeviceADD(uint256 index , bytes32 PSK, byes32 MAC );
    event RegisterDev(uint256 index , uint256 DID, bytes32 PSK );
    function RegisterDevice (uint256 DID, bytes32 PSK ) public {···
    function GetMAC (uint256 DID) private view returns(bytes32)  {
    function GetPSK (uint256 DID) private view returns(bytes32)  {
    function isRegistered(uint256  DIDn) private returns (bool) {···
    function isAuthorized(uint256 DIDn) public ···
    function AuthorizeDev(uint256 DID , bytes32 MAC ) public {···
    function Ifnull(bytes32 _valeur) public pure returns (bool) {···
```

Figure. 5.25. Contrat intelligent de la chaîne principale.

Lors du processus d'enregistrement du nœud, celui-ci génère d'abord ses clés privée et publique, puis envoie son identifiant (sa clé publique) et sa position (x, y, z) à la station de base. La station de base vérifie ensuite si le dispositif est pré-enregistré en appelant la fonction isRegistred() du contrat intelligent de la chaîne principale présentée dans la Figure.5.25.
Si le nœud est pré-enregistré, la fonction GenMac(DID , y), déployée sur la mainchain et présentée dans la Figure. 5.26, est appelée pour générer et stocker le MAC dans la mainchain.
Le code d'authentification de message sera généré d'une façon pseudo aléatoire comme suit : MAC = Hash(Y || RandomNumber), où Y est le contenu du message réçu .

```
// Mainchain Smart Contract
    function GenMAC(uint256 DID, bytes32 y) private {
    // Générer un nombre aléatoire
    uint256 randomNumber = uint256(keccak256(abi.encodePacked
    (block.timestamp, , msg.sender)));
    // Concaténer le nombre aléatoire avec 'y'
    bytes memory concatenated = abi.encodePacked(y, randomNumber);
    // Calculer le hash de la concaténation
    bytes32 mac = keccak256(concatenated);
    // Appeler la fonction AuthorizeDev avec le DID et le mac généré
    emit AuthorizeDev(DID, mac);
}
```

Figure. 5.26. Fonction Générer MAC

2.3.3.2. Phase d'authentification

Device.GenKeyPair(DevPrK, DevPbK)
Device.Z = Hash(MAC || PSK || DID || TSTP)
Device.Message1 = DID || Z || TSTP || DevPbK
Device.SendToBaseStation(Message1)
if !BS.MainChain.isAuthorized(DID) then
Return "Failed" // Authentication failed
else :
BS.MAC[DID] = MainChain.getMAC(DID)
BS.PSK'[DID] = MainChain.getPSK(DID)
if Hash(BS.MAC[DID] || BS.PSK'[DID] || DID || TSTP) != Z then
Return "Failed" // Authentication failed
else
BS.SideChain.AuthenticateDev(DID, BS.PublicKey, DevPbK)
*BS.SK[DID] = BS.PrivateKey * DevPbK*
BS.Message2 = Encrypt(SK || BS.PublicKey, DevPbK)
BS.SendToDevice(Message2)
Device.DecryptedMessage2 = Decrypt(Message2, DevPrivateKey)
*DevSK' = DevPrK * BS.PublicKey*
if DevSK' equals SK then Return "Successful"
else Return "Failed" // Authentication failed
end if , end if

Figure. 5.27. Pseudocode de la phase d'authentification

Les clés privées et publiques sont générées par le dispositif et la station de base, contribuant à établir une fondation de communication sécurisée. La cryptographie sur courbes elliptiques (ECC) dans le protocole d'échange de clés Diffie-Hellman sur courbes elliptiques (ECDH) [74] facilite la dérivation d'une clé de session (SK). De plus, une vérification des horodatages et du statut d'enregistrement des dispositifs renforce le système contre les menaces potentielles.
L'invocation du contrat intelligent de la fonction AuthenticateDev() dans la sidechain conserve les informations essentielles comme indiqué dans la Figure. 5.28, maintenant un historique complet des dispositifs authentifiés.

```
    function AuthenticateDev( bytes32 DPbk ,bytes32 BSPbk,
                              uint256 x,uint256 y,uint256 z) public {
        /* Store the Authenticated Device and the BS
        in the SideChain */
        require(msg.sender == address(MainChainContract));
        AuthenticatedDevice = Device(DPbk,BSPbk,x,y,z );
        emit AuthenticateDev(DPbk,BSPbk,x,y,z );
        AuthDevices[Dpbk] = AuthenticatedDevice; }
}
```

Figure. 5.28. Fonction de la sidechain : Authenticate Device

L'utilisation de la blockchain nous permet de suivre et de surveiller les dispositifs authentifiés, d'identifier la station de base responsable et d'horodater chaque événement d'authentification. Ces capacités de surveillance contribuent à l'analyse de la sécurité du réseau et fournissent des informations précieuses sur les performances globales du système.
À la fin de la phase d'authentification, la BS génère, pour chaque nœud :
1- Les identités de ses voisins et leurs distances (seuls les voisins directs, même s'ils sont dans un autre cluster).
2- La clé de session, qui est calculée via le système cryptographique ECDH [74] comme suit :
Les clés privées D_{PRK} et BS_{PRK} sont des entiers choisis aléatoirement dans l'intervalle [1, n - 1], où n est l'ordre de la courbe.
Les clés publiques de la BS et du dispositif sont corrélées avec leurs clés privées respectives.
Depuis :

$$D_{PBK} = D_{PRK} * G \qquad (3.1)$$

$$BS_{PBK} = BS_{PRK} * G \qquad (3.2)$$

où G est le point de base sur la courbe, défini par la station de base et n'importe quel dispositif pendant la phase de pré-enregistrement. La clé secrète partagée est donnée par :

$$SK = BS_{PRK} * D_{PBK} = BS_{PRK}*(D_{PRK} *G) = D_{PRK}* BS_{PRK} * G = D_{PRK} * BS_{PBK} \qquad (3.3)$$

Il est important de noter que le point de base G est prédéterminé et partagé entre tous les nœuds, car il est défini par la station de base lors de la phase de pré-enregistrement et transmis à chaque dispositif. Par conséquent, chaque nœud du réseau utilise le même point de

base G pour la génération de clés. Ainsi, l'équation (3.3) produit la clé secrète partagée, garantissant une communication sécurisée entre la station de base et le dispositif.

2.3.3.3. Phase de communication

Comme déjà mentionné, dans notre approche, la communication M2M peut être réalisée avec une infrastructure via une station de base (D2I/I2D) ou sans infrastructure en mode D2D.

Communication D2I : Pendant la communication D2I, lorsqu'un nœud cherche à communiquer avec une destination dans un autre réseau, éventuellement située loin de son cluster d'origine, il initie la transmission en envoyant des messages de demande de route RREQ (Route request en Anglais) aux nœuds voisins. Ces nœuds transmettent à leur tour les paquets au chef de cluster, qui les achemine ensuite vers la station de base correspondante pour les livrer à la destination finale. Ce processus s'applique également lorsque la station de base elle-même est la destination, comme lorsque les nœuds captent des données et les transmettent à la station de base. Dans les deux scénarios, le chemin de communication implique la transmission à travers les nœuds voisins, les chefs de cluster, et atteint finalement la station de base désignée pour un traitement ou un stockage ultérieur.

Communication D2D : La communication D2D permet aux dispositifs M2M d'échanger des données directement de manière ad hoc, contournant la station de base. Ce mode est efficace pour les transferts de données rapides et réduit la latence dans la communication.

2.3.3.4. Protocole de routage proposé

De manière similaire au protocole AODV, le schéma de routage proposé fonctionne en deux phases : la découverte du réseau et la transmission des données. Si un nœud source n'a pas de route vers la destination, il utilise un message de demande de route (RREQ) présenté dans la Figure. 5.29 pour découvrir un chemin vers la destination.

<table>
<tr><td colspan="2">SEEM-D2D Route Request</td></tr>
<tr><td colspan="2">AODV Route Request (192 Bits)</td></tr>
<tr><td>Encrypted Session Key (80 bits)</td><td>Average Distance(16 bits)</td></tr>
</table>

Figure. 5.29. Le message Route Request

Le message RREQ de SEEM-D2D contient, en plus des champs du message RREQ dans le protocole AODV, la distance moyenne , et la clé de session cryptée en utilisant la clé publique de la destination.
La distance moyenne AD (Average Distance en Anglais) d'un itinéraire peut être calculée à l'aide de l'équation suivante :

$$AD = [\sum_{i=1}^{N-1} D(i, i+1)] / (N-1) \qquad (3.4)$$

Où N est le nombre de nœuds formant l'itinéraire, D(i,j) représente la distance entre deux nœuds voisins i et j, et est calculée comme suit :

$$D(i,j) = \sqrt{(Xj - Xi)^2 + (Yj - Yi)^2} \qquad (3.5)$$

Où (Xi,Yi) et (Xj,Yj) représentent respectivement les coordonnées des nœuds i et j.
À partir de (3.4), lorsqu'un nœud reçoit un RREQ, il calcule de manière itérative la distance moyenne ADi en utilisant la formule (3.6). Ce processus itératif permet aux nœuds d'évaluer progressivement la distance moyenne du nœud source au nœud de destination à travers les nœuds intermédiaires.

$$AD_i = \frac{D(i-1,i) + (N-2)* ADi-1}{N-1} \text{ with } AD_0 = 0 \text{ , } i \in [1, N] \qquad (3.6)$$

À chaque itération, la distance moyenne calculée est insérée dans le paquet RREQ avant d'être transmise au nœud voisin suivant dans le réseau, jusqu'à ce que le RREQ atteigne le nœud de destination. En accumulant les informations de distance moyenne le long du chemin, les nœuds contribuent à construire une compréhension complète de la topologie du réseau, facilitant la détermination du chemin le plus efficace pour la transmission ultérieure des données.

Lorsque la destination reçoit le message RREQ, elle répond avec un message de réponse de route (RREP) illustré dans la Figure. 5.30 informant le nœud source du chemin à emprunter pour atteindre la destination.

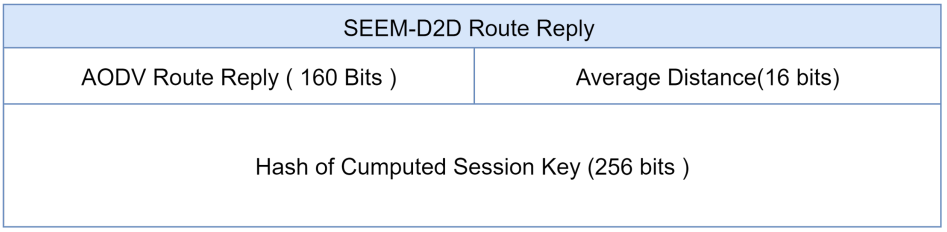


Figure. 5.30. Le message “Route Reply”

Le diagramme de flux représenté dans la Figure. 5.31 représente le processus de routage du protocole SEEM-D2D. Le processus commence lorsqu'un nœud source, appelé 'Source', souhaite communiquer avec un nœud de destination, appelé 'Destination'.

1- Initialisation : Avant de transmettre, le nœud source initialise le processus de routage en consultant sa propre table de routage pour vérifier si une route vers la destination est déjà connue. Si une telle route existe, le nœud source l'utilise, et le processus de découverte de route s'arrête.

2- Route inconnue : Si le nœud source n'a pas d'entrée pour la destination, il calcule la clé de session en utilisant l'équation (3.7) car les nœuds peuvent être identifiés par leurs clés publiques, puis génère un message de demande de route (RREQ) pour découvrir le chemin optimal en termes d'énergie requise pour atteindre la destination.

$$SK= Src_{PRK}*Des_{PBK}= Src_{PBK}* Des_{PRK} \qquad (3.7)$$

Le nœud source connaît ses voisins authentifiés car la station de base l'informe de leurs identités et positions à la fin de la phase d'authentification. Les tables de routage sont construites sur la base des informations extraites de la blockchain, garantissant que les voisins répertoriés sont authentifiés.

3- Découverte de la route: Tout d'abord, en cas de communication D2I, les données sont collectées par le nœud n pour être transmises à la BS via le CH. Si le nœud n n'a pas de route vers le CH, il envoie un message RREQ en multi saut au chef de cluster. Le chef de cluster renvoie un ou plusieurs RREP en fonction du nombre de RREQ reçus, indiquant le chemin vers la destination. Le nœud source n sélectionne le meilleur chemin en termes d'énergie et commence à envoyer les données collectées.
Deuxièmement, en cas de communication D2D, le schéma sélectionne un chemin sécurisé et optimal en termes de consommation d'énergie entre les appareils (ou machines) communicants. Le nœud n envoie en multidiffusion un RREQ à tous ses voisins, ce qui inclut son identifiant (clé publique), la distance moyenne initialisée à 0 et la clé de session chiffrée qui sera utilisée pour communiquer avec le nœud de destination. Cette dernière est calculée par le nœud source selon l'équation (3.7), chiffrée avec la clé publique du destinataire, puis insérée dans le message de demande de route.
À la réception d'un message RREQ, le nœud intermédiaire met à jour sa table de routage en ajoutant l'expéditeur du message RREQ, puis calcule la distance moyenne en utilisant la

formule (3.6) et diffuse le message RREQ à ses nœuds voisins jusqu'à ce que le message atteint la destination.
À la réception d'un message RREQ par la destination, celle-ci déchiffre le message en utilisant sa clé privée, calcule la clé de session et la compare à la clé de session incluse dans le message. Si les deux clés de session correspondent, le nœud de destination génère le message de réponse de route RREP, qui contiendra le hachage de la clé de session utilisant la clé de session. Ce message RREP est ensuite renvoyé au nœud source via le voisin qui a transmis le message RREQ.
Les nœuds intermédiaires jouent un rôle dans l'établissement de la route en ajoutant l'expéditeur du RREP (destination) à leurs tables de routage, puis en envoyant le message RREP au nœud source via le prochain saut stocké dans leurs tables de routage.
Une fois que le nœud source reçoit le message RREP, il déchiffre le message en utilisant la clé de session pour récupérer le hash. Ensuite, il calcule le hash de la clé de session et le compare au hachage reçu. Si le hachage calculé correspond au hachage reçu, l'intégrité de la route est vérifiée, ce qui indique que les deux parties possèdent la bonne clé de session. Cette vérification réussie leur permet de communiquer de manière sécurisée.

4- ***Mise en cache des routes :*** Le nœud source met en cache les routes reçues dans sa table de routage pour les communications futures, à la fois au sein de son cluster et entre les clusters. En tirant parti des mécanismes de mise en cache des routes, on garantit une utilisation efficace des routes précédemment découvertes, réduisant les surcharges et améliorant les performances globales du réseau.

5- Transfert de données : Une fois que les routes sont établies, le nœud source sélectionne le chemin optimal en fonction de facteurs tels que le nombre de sauts et la distance moyenne stockée dans sa table de routage avant d'initier le transfert de données. Cette route reste active tant qu'elle est viable. En cas de défaillance de liaison, le dernier nœud dans la route envoie un message d'erreur de route (RERR) au nœud source, indiquant que la destination est inaccessible, comme illustré dans la Figure. 5.32. Si le nœud source a toujours besoin d'une route vers la destination, ou s'il n'a pas d'autre route vers la destination, il doit initier à nouveau le processus de découverte de route.

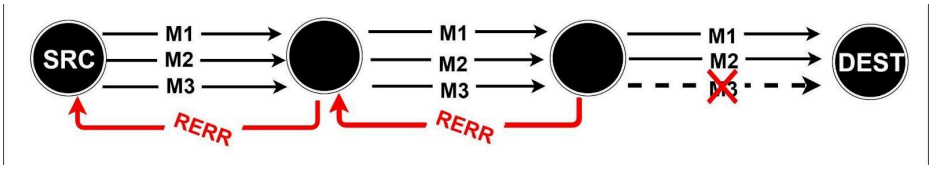


Figure. 5.32. Erreur d'itinéraire.

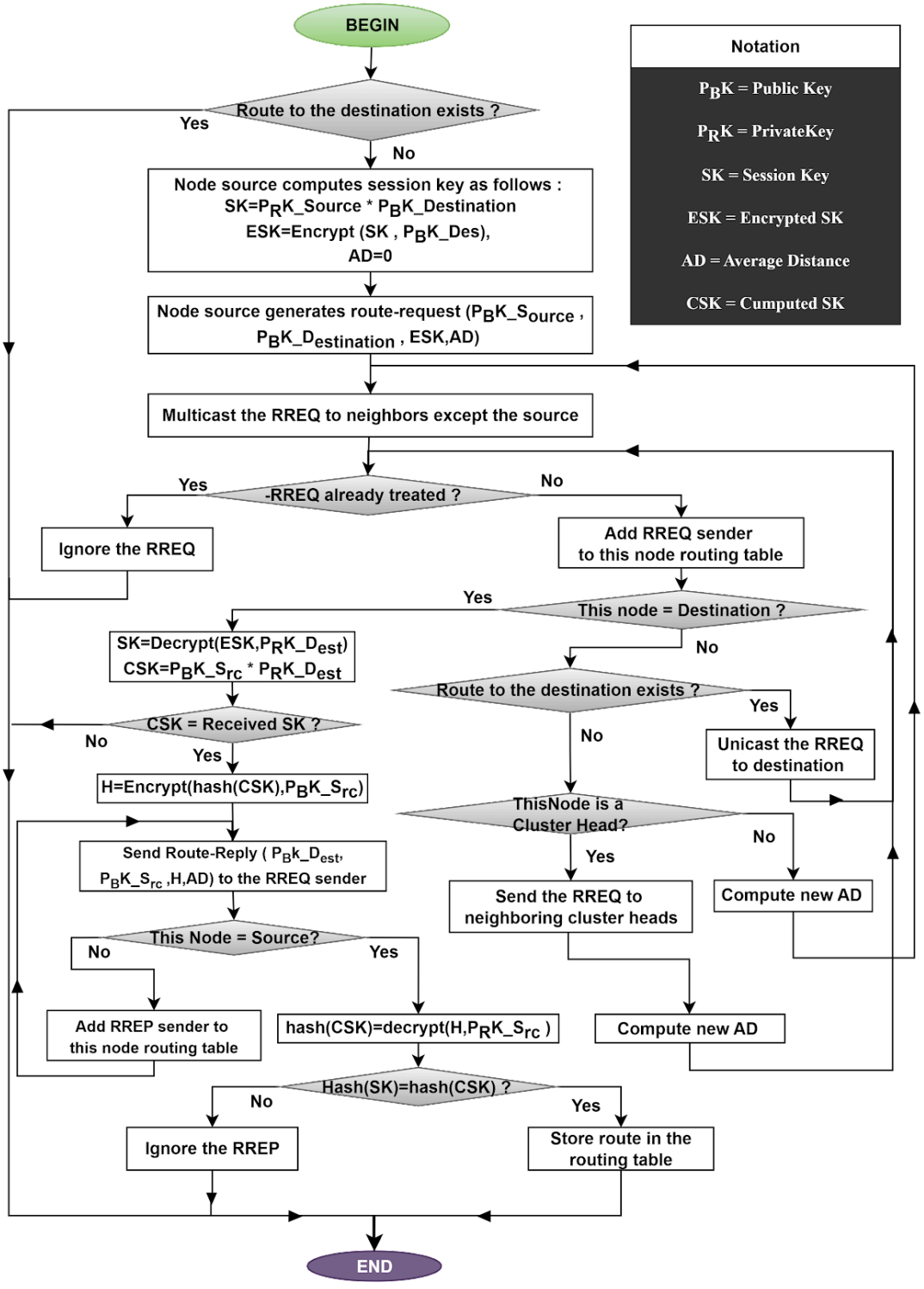


Figure. 5.31. Diagramme de flux du processus de routage

Pour élucider le système de communication adopté dans notre conception, trois scénarios différents illustrant la communication intra-cluster D2D, la communication inter-cluster D2D et la communication D2I (dispositif vers infrastructure) sont présentés dans la Figure. 5.33.

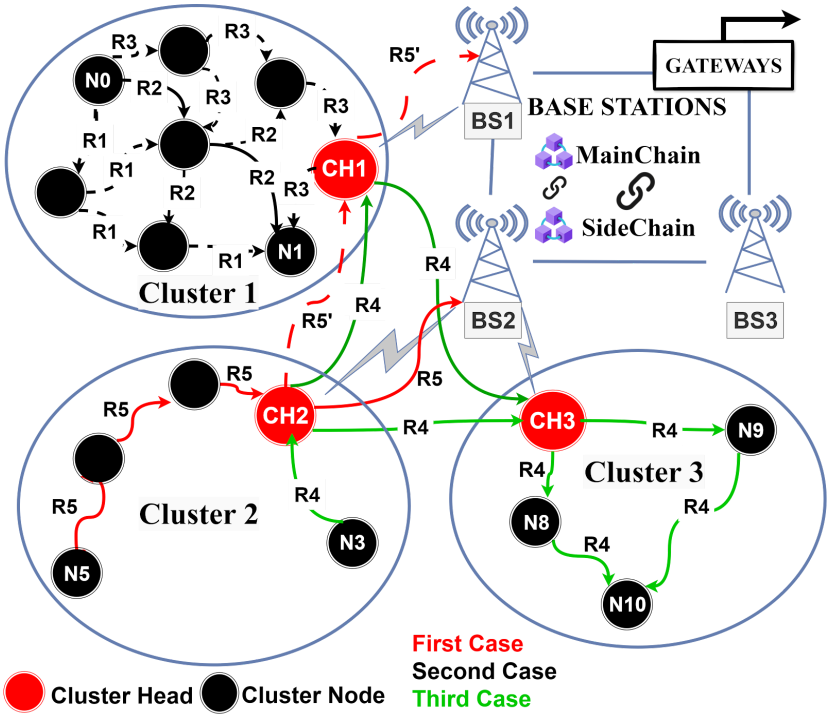


Figure. 5.33. Schéma de communication proposé.

Communication D2I: Ce scénario de communication se produit lorsque la destination est uniquement accessible via la station de base ou lorsque la station de base elle-même est le nœud de destination.

Un exemple est illustré dans le premier cas présenté dans la Figure. 5.33, où le nœud N5 capture des données dans le but de les transmettre à une station de base (BS2) pour traitement ou stockage dans la blockchain. La transmission des données suit l'itinéraire R5, qui comprend les nœuds voisins, CH2, et BS2.

Nous remarquons que le schéma de routage proposé démontre une tolérance aux pannes robuste, garantissant une communication ininterrompue même en cas de défaillance de la station de base. Par exemple, si BS2 rencontre une défaillance dans le premier cas illustré dans la Figure. 5.33, CH2 envoie les données à d'autres chefs de clusters. Dans une telle situation, les données sont routées vers CH1, qui les relaie ensuite à BS1 (route R5').

Communication intra-cluster D2D: Dans ce cas, les nœuds communiquent au sein de leurs clusters respectifs. Le chef de cluster n'a pas de tâche spéciale ; il joue simplement le rôle d'un nœud intermédiaire.

Par exemple, dans le deuxième cas illustré dans la Figure. 5.33, le nœud N0 envoie une RREQ à ses nœuds voisins, en spécifiant N1 comme destination. En conséquence, N1 reçoit trois RREQs le long des routes R1, R2 et R3. En réponse, N1 envoie trois RREPs. Le nœud N0 reçoit ensuite et stocke les trois routes dans sa table de routage.

Il est à noter que les routes R1 et R2 ne nécessitent pas l'intervention du chef de cluster et peuvent être utilisées pour la transmission de données. Même dans la route R3, le chef de cluster agit comme un nœud intermédiaire comme les autres nœuds au sein du cluster.

Communication inter-cluster D2D: Dans les scénarios où les nœuds source et destination ne se trouvent pas dans le même cluster (la destination est soit dans le même réseau soit dans

un autre réseau couvert par une autre station de base), si le nœud source envoie une RREQ et ne reçoit pas de RREP en réponse, le chef de cluster, agissant en tant que nœud relais spécialisé, transmettra la RREQ à d'autres chefs de cluster en utilisant le mode D2D. Ce processus se poursuit jusqu'à atteindre la destination prévue.

Par exemple, considérez le troisième cas illustré dans la Figure. 5.33, où N3 situé dans le cluster 2 souhaite communiquer avec N10 du cluster 3, la RREQ est routée de N3 à tous ses voisins dans le cluster jusqu'à ce qu'elle atteigne le chef de cluster (CH2). Ce dernier mettra à jour la RREQ (le champ de distance moyenne en utilisant la formule (6)), puis l'envoie à ses chefs de clusters voisins (CH1, CH3). CH3 dans ce cas sera responsable de router le paquet vers N10, et une fois que le nœud de destination reçoit la RREQ, il répond avec un RREP via le chemin utilisé. Le nœud N3 capture les informations de routage et les stocke dans sa table de routage telles que le nœud de destination, le nœud suivant, le nombre de sauts et la distance moyenne jusqu'à ce nœud, comme illustré dans la Figure. 5.34.

Routing Table				
Seq	**Destination**	**Next hop**	**Hop**	**Average Distance**
1	BS2	CH2	2	AD(N3,BS2)
2	CH2	CH2	1	AD(N3,CH2)
2	N10	CH2	4	AD(N3,N10)

Figure. 5.34. La table de routage du dispositif N3

2.3.4. Évaluation des performances

Dans cette section, nous évaluons les performances du mécanisme d'authentification et du protocole de routage proposés, en termes de paramètres clés tels que les surcharges de communication et de calcul, le délai moyen et la consommation d'énergie introduits par le système proposé.
À des fins de comparaison, cette section traite des protocoles liés au routage sécurisé, à savoir ACSRP[162], SAODV[163], IJS[164][165][166][167][168]. Ces protocoles fournissent une authentification et une sécurité pour les communications multi-sauts dans les réseaux sans fil. Une comparaison de leurs performances avec celles du SEEM-D2D proposé est présentée ci-dessous.

Un protocole de routage sécurisé basé sur AODV pour les réseaux IoT sans fil multi-saut est proposé dans [162] . Dans ce protocole, chaque nœud intermédiaire signe le paquet à relayer en utilisant une signature basée sur l'identification. Chaque nœud génère sa propre RREP signée pour empêcher toute altération de paquets et les attaques par des nœuds malveillants. Enregistrer les réponses de routage dans la table de routage permet à un nœud intermédiaire de prouver la validité du chemin à une tierce partie et de générer des RREPs pour répondre aux RREQs avant la destination dans le processus de routage.

Le protocole Secure Ad hoc On-Demand Distance Vector (SAODV) [163] aborde le problème de la sécurisation d'un réseau MANET. SAODV est une extension du protocole de routage AODV qui peut être utilisée pour protéger le mécanisme de découverte de route en fournissant des fonctionnalités de sécurité telles que l'intégrité, l'authentification et la non-répudiation. SAODV suppose que chaque nœud ad hoc dispose d'une paire de clés de signature d'un système de cryptographie asymétrique approprié. De plus, chaque nœud ad hoc est capable de vérifier de manière sécurisée l'association entre l'adresse d'un nœud ad hoc donné et la clé publique de ce nœud. La gestion de clés est chargée de réaliser cela.

Un mécanisme d'authentification de nœud multi-saut est adopté par le protocole Indirect Join to the Sink (IJS), introduit dans [164], qui présente une approche innovante pour permettre aux nouveaux nœuds de rejoindre le réseau via des nœuds intermédiaires, atteignant finalement le puits pour l'authentification. IJS propose différentes versions, chacune adaptée à des contraintes et des besoins de sécurité spécifiques.

Dans [165], les auteurs proposent un nouveau protocole de routage pour les réseaux cellulaires multihop (MCN). Ce protocole sélectionne un chemin sécurisé et court dans le processus de communication. Il assure la confidentialité en utilisant le schéma Well Pairing et applique le schéma Smart-Chen-Kudla pour la génération de clés et l'authentification. Il fournit également l'anonymat, où les nœuds participants génèrent des identités temporaires pour chaque session de communication afin d'éviter de révéler leur identité réelle.

Dans [166], Lutful Karim et al. présentent un schéma de clustering tolérant aux fautes, économe en énergie et sécurisé pour les réseaux de zone M2M. Le mécanisme de sécurité

proposé génère et attribue des clés secrètes aux dispositifs M2M. Il utilise la cryptographie à clé symétrique à deux niveaux où chaque paire de dispositifs utilise la même clé secrète pour transmettre des données. Les clés partagées basées sur la permutation sont utilisées entre les nœuds membres, les CH (Cluster Heads) et la passerelle ; avant de transmettre des données. La clé partagée ou la permutation simple est utilisée à la fois pour chiffrer et déchiffrer les données réelles. Tant la clé secrète que les données captées sont chiffrées à l'aide du chiffre de Vigenère.

Dans [167], les auteurs proposent un cadre appelé SEMUD pour sécuriser les communications D2D multi-saut contre les attaques de déni de service (DoS) et autres, dans les réseaux sans fil avec et sans infrastructure cellulaire. SEMUD propose pour les applications D2D une gestion des certificats pour l'authentification mutuelle des utilisateurs et la découverte des voisins pour le service de disponibilité. De plus, SEMUD utilise la cryptographie à clé symétrique pour la confidentialité et l'intégrité des messages.
Dans [168], un protocole de routage multi-saut sécurisé pour les réseaux sans fil multi-saut est proposé pour sélectionner un chemin sécurisé et optimal pour la communication de données. Le protocole assure des services de confidentialité et d'authenticité basés sur une méthode d'accord de clés utilisant le schéma Well Pairing. Le service d'intégrité est réalisé par la vérification de la fonction MAC.

2.3.4.1. Surcharge de calcul

Nous avons comparé la surcharge de calcul pour la découverte de route de notre protocole SEEM-D2D et du protocole IJS, pour un nombre N de sauts. Nous avons calculé le coût de calcul de chaque protocole et résumé les résultats dans la Table 5.7.

Table 5.7 : Le coût de la surcharge de calcul du routage

Protocol	Routing Computation Overhead Cost
SEEM-D2D	2*T(Asym)+2*T(Sym)+2*T(Eca)+ 2*T(hash)
IJS[164]	2*T(Asym) + 2*N*T(Sym)

Les nœuds intermédiaires dans le protocole SEEM-D2D se contentent de transmettre les messages au prochain saut sans effectuer de calculs. En revanche, IJS demande aux nœuds intermédiaires de déchiffrer ou de chiffrer les messages à chaque saut, ce qui entraîne une complexité de O(N). Avec notre protocole démontrant une complexité computationnelle de O(1), il s'avère être une option plus efficace et supérieure par rapport à IJS, présentant des coûts de calcul plus faibles et le rendant ainsi l'option préférée pour une implémentation pratique.
Le graphique représenté dans la Figure. 5.35. indique que notre protocole proposé surpasse systématiquement le protocole IJS à mesure que le nombre de nœuds intermédiaires augmente en termes de temps de calcul.

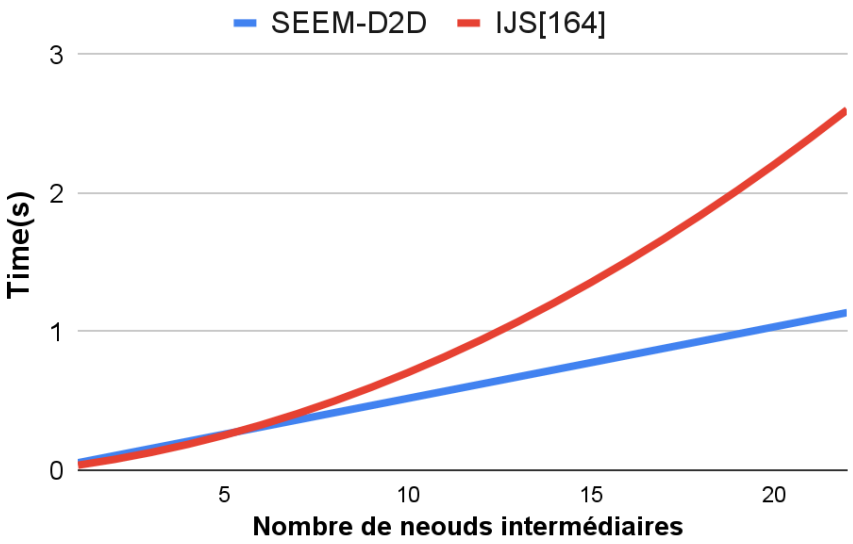


Figure. 5.35. Comparaison de la surcharge de calcul

2.3.4.2. Surcharge de communication

La surcharge de communication (Communication overhead en Anglais) est déterminée par la longueur totale des champs ajoutés dans les messages transmis et reçus lors du processus d'établissement de la route, mesurée en octets. Dans cette phase, le protocole utilise deux messages, RREQ pour la demande de route et RREP pour la réponse de route. La surcharge de communication introduite par le processus de sécurité peut être calculée comme suit :
Surcharge de communication= RREQ (11 bytes) + RREP (34 bytes),
Où : Surcharge de RREQ = Session Key (10 bytes) + Average Distance (2 bytes) ,
Surcharge de RREP = Hash(CSK) (32 bytes) + Distance moyenne (2 bytes).

Table 5.8 : Coût de surcharge de communication (CSC) pour N sauts.

Protocol	SEEM-D2D	SAODV[163]	IJS[164]
RREQ Longueur	12 Bytes	308 Bytes	(4N + 5) Bytes
RREQ CSC	12 * N	(N+1) * 308	$\sum_{0}^{N}(4N + 5)$
RREP Longueur	34 Bytes	(N + 308) Bytes	(4N+5) Bytes
RREP CSC	34*N	(N+1) * 308	$\sum_{0}^{N}(4N + 5)$
Total CSC	46*N	616*N+616	$2\sum_{0}^{N}(4N + 5)$
Complexité	O(N)	**O(N)**	**O(N²)**

Le tableau 5.8. fournit une comparaison des surcharges de communication pour divers protocoles. Notamment, le protocole proposé présente des exigences de stockage réduites et des coûts de communication inférieurs par rapport à d'autres protocoles alternatifs.

En termes de surcharge de communication, le protocole proposé présente une complexité de O(N) tout comme SAODV[163]. Cependant, le protocole SEEM-D2D surpasse SAODV et IJS [164] en raison de sa taille de message significativement plus petite de 42 octets par rapport aux 616 octets de SAODV et à la complexité de $O(N^2)$ d'IJS.

2.3.4.3. Consommation énergétique

Pour évaluer la consommation énergétique des appareils, nous définissons l'énergie totale consommée (Etot), calculons l'énergie utilisée lors de la transmission de la réception en fonction de la taille du paquet de données (L) et de la distance de transmission D(i, j) entre deux nœuds i et j. L'énergie totale Etot utilisée pour authentifier N dispositifs dans notre scénario peut être estimée en utilisant le modèle de communication radio du premier ordre [156] comme suit :

$$Etot = \sum_{j=1}^{N} \left(\sum_{i=1}^{M} Li * Eelec + Li * Eamp * D(j,i)^2 + \sum_{i=1}^{M'} Li * Eelec \right) \quad (3.8)$$

où :*Li*: La taille totale en bits des données transmises par le dispositif i.
E_{ELEC} : la dissipation d'énergie radio.
E_{AMP} : la dissipation d'énergie de l'amplificateur de transmission.
N est le nombre de nœuds, M est le nombre de messages émis, M' est le nombre de messages reçus

2.3.5. Résultats de la simulation

Table 5.9 : Paramètres de simulation

Paramètre	Description
Platforme	NS3.33/Ubuntu 20.04
Programmation	C++
Technologie radio	802.15.4
Nombre de nœuds	200
Puissance de transmission (dBm)	0 dBm
Eamp	100pJ/bit/m²
Eelec	50 nJ/bit

Pour analyser la consommation d'énergie et les temps de transaction pour l'établissement de routes dans les réseaux M2M à saut unique et à plusieurs sauts, nous faisons varier la distance moyenne entre les nœuds participants dans notre simulation. Ce paramétrage de la distance offre des informations précieuses sur l'efficacité et la praticité de différentes stratégies de routage dans divers scénarios de réseau. De plus, notre scénario de simulation comprend trois

clusters avec une station de base positionnée au centre de la zone du réseau, mesurant 200m x 200m.

L'utilisation d'énergie pour les dispositifs de transmission et de réception est quantifiée à 50 nJ/bit, tandis que la consommation d'énergie de l'amplificateur de transmission est de 100 pJ/bit/m². Il est important de noter que les autres paramètres sont supposés conserver leurs configurations par défaut telles que spécifiées dans NS3. Les paramètres de simulation sont présentés dans la Table 5.9.

La distance moyenne entre les nœuds est cruciale car elle impacte directement l'énergie dépensée lors de l'établissement de la route. À partir de la formule (3.8), des distances plus longues entraînent généralement des coûts énergétiques plus élevés pour la transmission et le relais de données. En incorporant ce calcul dans notre analyse de la consommation d'énergie, nous pouvons mieux comprendre les compromis entre la distance, la consommation d'énergie et l'efficacité de la route dans les réseaux M2M à un ou plusieurs sauts.

La simulation explore divers scénarios de communication, allant d'un à huit sauts pour atteindre une destination à des distances variables. Les résultats révèlent que différents nombres de sauts sont optimaux pour différentes gammes de distances. Par exemple, un saut est plus favorable que la communication multi-saut pour les distances plus courtes, tandis que deux, quatre, voire huit sauts deviennent plus économes en énergie pour les distances plus longues. Cette observation souligne l'adaptabilité de notre protocole de routage sécurisé, permettant aux dispositifs de sélectionner la route la plus économe en énergie en fonction de leurs besoins de communication spécifiques.

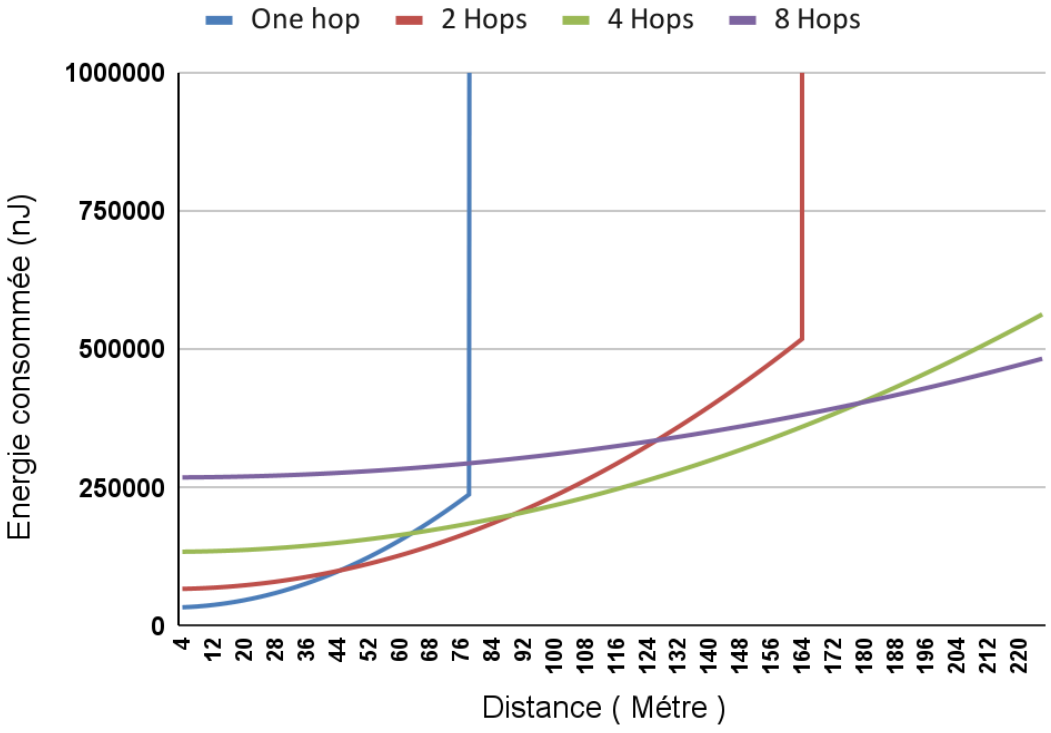


Figure. 5.36. Consommation d'énergie par type de communication à différentes distances

Figure. 5.36. illustre la consommation d'énergie de SEEM-D2D pour des distances variables dans les scénarios de routage à un seul saut et à plusieurs sauts. Le diagramme présente quatre courbes représentant la consommation d'énergie pour SEEM-D2D avec des scénarios de routage à un saut, deux sauts, quatre sauts et huit sauts. Les observations révèlent que pour

des distances inférieures à 20 mètres, le routage à un seul saut représente le protocole le plus économe en énergie. À mesure que les distances dépassent 20 mètres, le routage à plusieurs sauts devient plus économe en énergie.

Cependant, à la distance de croisement critique de 80 mètres, la consommation d'énergie du routage à un seul saut augmente significativement en raison des limitations du protocole 802.15.4, qui ne peut pas transmettre au-delà de cette distance. Une tendance similaire est observée pour le routage à deux sauts à 160 mètres. La Table 5.10 fournit un classement des types de communication en fonction de la distance moyenne entre la source et la destination.

Ce classement illustre le type de communication préféré en termes de nombre de sauts, pour différentes plages de distance moyenne entre les nœuds du réseau.

Table 5.10 : Classement par type de communication à des distances variables.

Distance	Meilleur	Deuxième	Troisième	Quatrième
0<D<45	One Hop	2 Hops	4 Hops	8 Hops
45<D<65	2 Hops	One Hop	4 Hops	8 Hops
65<D<80	2 Hops	4 Hops	One Hop	8 Hops
80<D<90	2 Hops	4 Hops	8 Hops	/
90<D<130	4 Hops	2 Hops	8 Hops	/
130<D<160	4 Hops	8 Hops	2 Hops	/
160<D<190	4 Hops	8 Hops	/	/
190<D	8 Hops	4 Hops	/	/

Le graphique présenté dans la Figure. 5.37 montre les résultats obtenus en termes de consommation d'énergie en microjoules [μJ] de quatre protocoles : SEEM-D2D, [162], [163] et [164] pour l'établissement de la route. La Figure. offre une étude comparative à travers un nombre variable de nœuds intermédiaires avec une distance moyenne fixée à 12 mètres.
Malgré la taille fixe du message de 308 octets dans le protocole SAODV, il reste significativement plus grand par rapport aux autres protocoles. Cette taille de message substantielle explique la consommation d'énergie élevée observée dans ce protocole.
En ce qui concerne le protocole IJS, sa consommation d'énergie est élevée en raison de l'inclusion d'informations supplémentaires avec un surcoût de 4 octets à chaque saut. Par conséquent, à mesure que le nombre de sauts augmente, la taille du message augmente également. Cette augmentation de la taille du message suit une séquence arithmétique, représentée spécifiquement par la formule (4N)+5. Cela résulte en une taille de message variable, où dans le X-ième saut, la taille du message échangé est donnée par (4X) + 5. Par exemple, dans le 20ème saut, la taille du message sera de 85 octets.

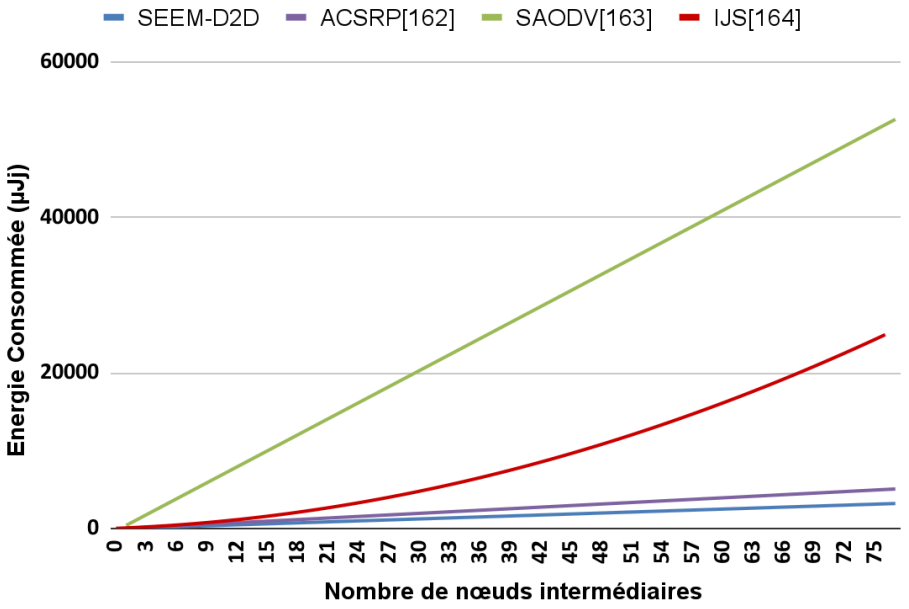


Figure. 5.37. Energie consommée pour l'établissement de la route vs nombre de sauts

D'autre part, le protocole présenté dans [162] montre une consommation d'énergie plus faible car la taille des messages échangés reste fixée à 320 octets. Le protocole SAODV[163] entraîne une consommation d'énergie élevée en raison de son important surcoût, tandis que le protocole IJS[164] connaît une consommation d'énergie croissante avec le nombre de sauts en raison de l'augmentation de la taille des messages. Le protocole proposé ainsi que le protocole de [162] démontrent une consommation d'énergie relativement faible, tandis que le protocole SEEM-D2D montre un bon niveau de sécurité et une efficacité énergétique en employant une authentification des nœuds voisins basée sur la blockchain.

La Figure. 5.38. illustre la surcharge énergétique introduite par le protocole SEEM-D2D à des fins de sécurité. Cette surcharge est principalement due au processus d'établissement de la route, où une clé de session de 80 bits, la distance moyenne de 8 bits, sont insérés dans le message de demande de route, et le hachage de la route empruntée par la demande de route, d'une taille de 256 bits, qui est inclus dans le message de réponse de route. Ces champs supplémentaires contribuent à l'augmentation de la consommation d'énergie observée dans le système SEEM-D2D avec des mesures de sécurité.

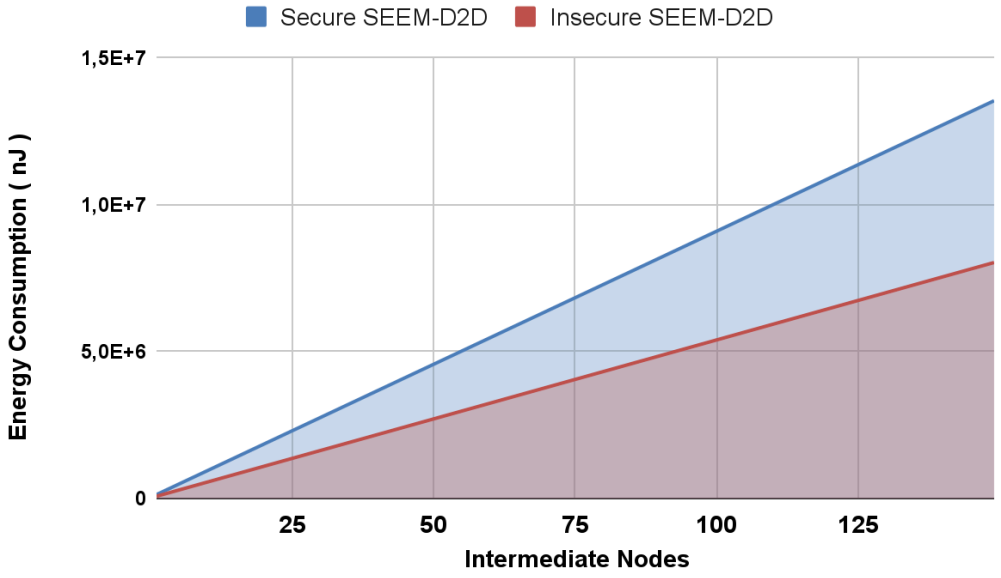


Figure. 5.38 Consommation d'énergie de l'établissement de route avec et sans sécurité.

Notre approche, telle qu'illustrée dans la Figure. 5.39, montre un impact minimal sur les temps de transaction d'authentification même avec l'augmentation du nombre de nœuds. Notamment, le protocole basé sur la sidechain dépasse systématiquement les modèles traditionnels de blockchain, présentant des temps de transaction plus courts. En incorporant une sidechain dans notre architecture réseau, nous adressons efficacement les défis de scalabilité couramment rencontrés dans les blockchains a une seule couche.

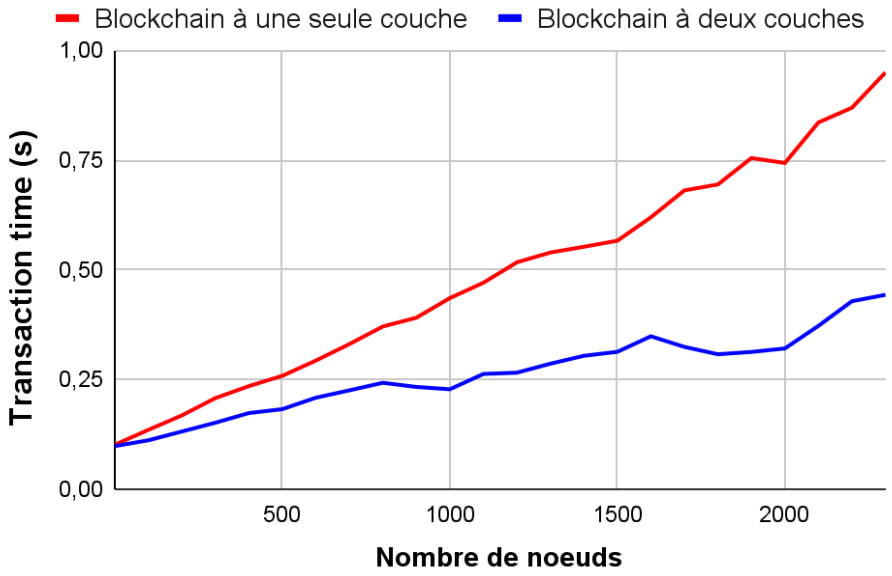


Figure. 5.39. Temps d'authentification en fonction du nombre de nœuds.

2.3.6. Analyse de sécurité

Notre schéma de sécurité intègre plusieurs mesures pour garantir une communication sécurisée. Tout d'abord, il assure une authentification mutuelle entre les dispositifs et la station de base. De plus, des fonctions de hachage sont utilisées à toutes les phases, servant de preuve de la validité de l'identité du dispositif.
Nous avons choisi AVISPA[10], un outil de vérification de sécurité reconnu, utilisant le langage de spécification de protocole de haut niveau (HLPSL) via le framework AVIPSA pour des descriptions détaillées de protocoles et leur traduction en un format intermédiaire (IF). L'évaluation de notre protocole impliquera une simulation avec le Model-Checker On-the-fly (OFMC).
Cette vérification évalue la résistance du protocole aux menaces actives, y compris les attaques de rejeu et de l'homme du milieu, confirmant l'authentification mutuelle et la confidentialité des clés privées. La spécification des propriétés de sécurité basée sur le HLPSL, comme illustré dans la Figure. 5.40., englobe l'assurance de la confidentialité du code d'authentification des messages (secrecy_MAC), la confidentialité de la clé de session Sk (secrecy_SK), et l'authentification mutuelle entre le dispositif et la station de base (authentication_on sk_auth_1).

```
goal
        secrecy_of secrecy_MAC,secrecy_SK
        %%for Mac, SessionKey secrecy
        authentication_on sk_auth_1
        %% Mutual Authentication
end goal
```

Figure. 5.40. Objectifs et propriétés de sécurité

2.3.6.1. Attaque de rejeu

Pour contrer une telle attaque pendant le processus d'authentification avec la station de base, notre protocole utilise un horodatage. Si un adversaire tente de rejouer une demande d'authentification avec un horodatage expiré ou précédent, la station de base rejette la demande. Cependant, si un adversaire tente de rejouer une RREQ, un numéro de séquence est utilisé pour vérifier si la demande a déjà été traitée à chaque saut. Si la Route-Request a déjà été traitée, elle sera ignorée.
Pour renforcer la sécurité et contrer les attaques par rejeu dans la phase de communication, nous avons opté pour l'utilisation d'un hash de la clé de session dans le message de réponse de route (RREP) au lieu de la clé de session cryptée. En effet, le message de demande de route (RREQ) contient la valeur de la clé de session cryptée. Si un attaquant parvient à intercepter cette valeur, il pourrait potentiellement l'utiliser pour générer une fausse RREQ. En utilisant un hash de la clé de session, nous nous assurons que même si la clé cryptée est interceptée, elle ne pourra pas être directement utilisée pour rejouer d'anciens messages. Le hash de la clé de session dans le RREP fournit donc une couche supplémentaire de sécurité, garantissant que seules les parties légitimes ayant connaissance de la clé peuvent valider et répondre aux demandes de route de manière sécurisée.

En ce qui concerne les attaques de rejeu, OFMC évalue si une partie légitime pourrait agir involontairement comme un intrus passif en exécutant un schéma spécifique. Par la suite, OFMC fournit à cet intrus des informations sur les sessions régulières menées par des agents légitimes. De plus, OFMC évalue le potentiel de l'intrus à mener une attaque de l'homme du milieu.
Sur la Figure. 5.41, deux sessions identiques ont été intentionnellement configurées pour simuler l'occurrence d'une attaque de rejeu. La spécification fournie ci-dessous pour le rôle de l'environnement dans HLPSL décrit comment ces deux sessions identiques entre le périphérique et la station de base sont traitées, permettant la détection des attaques de rejeu si elles se produisent.

```
intruder_knowledge= { source, destination,srcpublickey, destpublickey}

composition
session(source,destination,s1,srcpublickey,destpublickey,sk)
/\ session(source,destination,s1,srcpublickey,destpublickey,sk)
 %% Session2 For Replay Attack
/\ session(i,destination,s1,srcpublickey,destpublickey,sk)
%% Session3 For the intruder in the middle
```

Figure. 5.41. Les principales spécifications.

2.3.6.2. Attaque de l'Homme du Milieu (MIM)

Pour les communications D2I, lorsqu'une demande d'authentification est reçue par la station de base, celle-ci récupère les informations du périphérique depuis la blockchain et les hache, comparant le code de hachage résultant au code de hachage reçu pour détecter toute modification effectuée par une éventuelle attaque de l'homme du milieu.
Pour maintenir la sécurité du réseau dans les communications D2D, chaque nœud vérifie si les données reçues proviennent d'un expéditeur connu et de confiance en vérifiant sa table de routage qui contient les nœuds authentifiés. Si l'expéditeur n'est pas authentifié, tous les paquets entrants en provenance de celui-ci sont immédiatement rejetés. De plus, chaque nœud est identifié par sa clé publique, et avant que la communication ne se produise, le nœud source chiffre le message avec la clé de session calculée via ECDH [74], garantissant que seul le nœud de destination peut le déchiffrer.
La troisième session, commentée dans le rôle de l'environnement dans la Figure. 5.41, est conçue pour détecter la présence d'une attaque de l'homme du milieu, si elle se produit. Les résultats présentés dans la Figure. 5.42. fournissent des preuves que notre protocole est en effet sécurisé contre les attaques de l'homme du milieu, et il assure une authentification mutuelle.

2.3.6.3. Attaque d'usurpation d'identité

L'utilisation des clés publiques comme identifiants pour chaque nœud peut renforcer la sécurité de notre réseau. Si un nœud B tente de se faire passer pour un nœud A, il doit utiliser sa propre clé publique. Par conséquent, les réponses à ses messages seront toujours chiffrées, et il ne pourra pas les déchiffrer sans obtenir la clé privée du nœud A. Cela peut prévenir les attaques d'usurpation d'identité et protéger l'intégrité et la confidentialité des données de notre réseau M2M.

2.3.6.4. Attaque par déni de service distribué (DDoS)

Le protocole proposé peut atténuer les attaques DDoS en veillant à ce que les nœuds ne communiquent qu'avec des nœuds authentifiés. De plus, à chaque saut pendant la communication, les nœuds vérifient si la demande a déjà été traitée COMMENT?. Si c'est le cas, elle sera ignorée pour éviter un trafic inutile et réduire le risque d'attaques DDoS. Ce mécanisme peut aider à empêcher les nœuds malveillants de submerger le réseau avec des demandes répétées et garantir que les demandes légitimes sont traitées efficacement.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/SEEM_D2D.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.64s
  visitedNodes: 993 nodes
  depth: 12 plies
```

Figure. 5.42. Résultats de vérification du protocole

2.3.6.5. Authentification mutuelle

Une authentification mutuelle entre les nœuds et la station de base peut renforcer la sécurité de notre réseau. À la fin de la phase d'authentification pour chaque nœud dans le réseau, la station de base envoie au nœud sa liste de voisins (table de routage) et lui envoie également la clé PsK stockée dans la blockchain. Le dispositif vérifie alors l'exactitude de la PsK' pour authentifier une station de base. Cela garantit que chaque nœud ne peut communiquer qu'avec des nœuds déjà authentifiés dans le réseau. De plus, l'utilisation d'une table de routage sécurisée et de clés de session générées via ECDH [74] entre deux appareils avant toute communication D2D peut également renforcer la sécurité du réseau en garantissant que seuls les nœuds authentifiés peuvent communiquer entre eux, assurant ainsi que leur communication est chiffrée et sécurisée.

2.3.7. Discussion

Le protocole proposé offre des solutions efficaces pour des communications M2M sécurisées, comprenant l'authentification, la confidentialité, le clustering efficace du réseau et le support des paradigmes de communication à saut unique et multi-saut, ainsi que la communication de dispositif à dispositif (D2D) comme indiqué dans la Table 5.10 Cependant, il est conçu pour des réseaux de zones de dispositifs M2M homogènes avec une faible mobilité, ce qui peut limiter son applicabilité dans des environnements hautement dynamiques.
L'avenir des communications en réseau accordera une place plus importante aux communications M2M sans intervention humaine. La tendance va vers la coexistence de différents réseaux hétérogènes utilisant une variété de technologies de communication impliquant des appareils avec des capacités de traitement et de stockage variables. Cela encourage à attribuer des tâches distinctes aux appareils connectés (ou machines) dans la gestion et la sécurité des communications.

Table 5.11 : Caractéristiques de certains protocoles de routage M2M existants.

	AS	**CS**	**CL**	**SH**	**MH**	**D2D**
SAODV[163]	X	X	X		X	
IJS[164]	X	X			X	
[165]	X	X	X		X	
[166]	X	X	X		X	
[167]	X	X		X		X
[168]	X	X	X		X	
SEEM-D2D	X	X	X	X	X	X

AS: Authentication Service, CS: Confidentiality Service, CL: Clustering, SH: Single Hop, MH: Multi-hop, D2D: Device to Device

2.3.8. Conclusion

Cette contribution présente un protocole de communication multi-sauts économe en énergie et sécurisé basé sur une blockchain à deux couches. Les techniques de clustering, de D2D et de multi-sauts utilisant les nœuds voisins avec la réserve d'énergie la plus élevée fournissent une efficacité énergétique basée sur un routage léger à l'intérieur des clusters de dispositifs. Les expériences de simulation démontrent que la proposition introduit des surcharges d'énergie, de stockage et de retard limitées par rapport à d'autres schémas existants. De plus, la mise en œuvre d'une blockchain à deux couches permet une amélioration en termes de scalabilité du réseau, ce qui est un inconvénient associé aux blockchains à une seule couche. La sidechain introduite sert de mécanisme crucial pour décharger le traitement des transactions, permettant au réseau de s'adapter de manière transparente et de répondre aux demandes évolutives tout en maintenant des performances et une sécurité optimales.

En perspective à ce travail, nous envisageons dans des travaux futurs, d'élargir le champ d’application de notre étude aux réseaux M2M à haute mobilité tels que les réseaux véhiculaires.

CONCLUSION GENERALE

La sécurité revêt une importance capitale dans les communications M2M en raison de l'étendue de leur champ d'application, et de leur impact sur notre vie quotidienne. La sécurisation des systèmes M2M fait face à des défis posés par le déploiement étendu, la décentralisation et l'hétérogénéité des appareils. En outre, les solutions de sécurité classiques se heurtent à des difficultés lors de leur implémentation sur les dispositifs M2M qui sont généralement limités en ressources de mémoire, de traitement et d'énergie .

Cette thèse s'est intéressée à différents aspects de l'architecture M2M (caractéristiques, contraintes, etc.), comme défini par L'ETSI, en expliquant ses trois domaines (le domaine Application, le domaine Réseau de Communication, et le domaine Réseau des Dispositifs M2M). La normalisation des systèmes M2M a été abordée pour souligner l'intérêt que les organismes et alliances de standardisation accordent à ce créneau, pour éviter leur diversification et permettre le développement du marché des communications M2M. Certaines caractéristiques des communications M2M importantes (faible mobilité, faible consommation d'énergie, nature du trafic, tolérance temporelle,...) ont été mises en évidence, ainsi que les défis en matière de scalabilité, d'hétérogénéité, de congestion, de qualité de service, etc. qui ont été abordés en insistant en particulier sur le volet de la sécurité des communications M2M (Machine à Machine). Notre étude de l'état de l'art à ce sujet, nous a permis de dégager les vulnérabilités et les menaces à différents niveaux de l'architecture des systèmes M2M. Les attaques possibles contre ces systèmes ont été catégorisées et énumérées, ainsi que les architectures et solutions de sécurité existantes. Les nouvelles tendances pour la sécurisation des réseaux des dispositifs M2M, nous ont amené à explorer le domaine de la blockchain et ses potentialités à répondre à notre problématique. Nous avons examiné les concepts fondamentaux de la technologie blockchain, ses protocoles de consensus, ses types et ses applications dans le contexte de l'IoT/M2M.

Les solutions que nous avons proposées dans le cadre de cette thèse, s'articulent autour d'une architecture basée sur la blockchain placée sur une infrastructure Edge, à la périphérie du domaine réseau des dispositifs M2M (M2M Device Area Network)..

Dans notre première contribution, nous avons introduit un mécanisme d'authentification basé sur la technologie blockchain pour assurer les services d'autorisation d'accès, d'authentification et d'intégrité, en offrant de bonnes performances et des coûts réduits de traitement et de communication. En utilisant des clés cryptographiques plus courtes et exploitant les avantages de la technologie blockchain, notre proposition garantit une authentification plus puissante et un partage sécurisé des données tout en préservant leur intégrité, leur disponibilité et leur traçabilité. Contrairement aux protocoles traditionnels qui reposent sur une architecture centralisée, notre approche décentralisée élimine le besoin de gérer la confiance entre les appareils et les stations de base grâce à une entité centrale de certification, offrant ainsi un niveau de confiance élevé sans SPF (Single Point of Failure). Cependant, il convient de noter que l'implémentation de la blockchain entraîne une

augmentation du temps de traitement et de la consommation d'énergie, en particulier au niveau des stations de base.

Notre deuxième contribution a consisté en un protocole d'authentification basé blockchain, mais intégrant en plus, une deuxième blockchain latérale, pour résoudre le défi de scalabilité qui représente un handicap associé aux blockchains à une couche. La mise en œuvre d'une blockchain à deux couches permet d'améliorer l'évolutivité du réseau car la sidechain introduite sert de mécanisme pour décharger la blockchain principale du traitement des transactions, ce qui permet au réseau de mieux s'adapter à l'évolution, en maintenant des performances et une sécurité optimales.

Enfin, une troisième contribution a introduit un protocole de communication sécurisé et reposant sur une blockchain à deux couches, à travers une communication multi-sauts efficace en énergie, et utilisant la technologie D2D (Device to Device), qui représente un nouveau paradigme introduit dans les réseaux 5G et 6G, pour des communications directes sans passer par les infrastructures des réseaux cellulaires.
Bien que notre proposition offre des solutions efficaces pour sécuriser les communications M2M, il reste qu'elle est conçue pour des réseaux à faible mobilité, et présente donc des défis potentiels à relever pour son application dans des environnements hautement dynamiques et hétérogènes.
En tant que travaux futurs, et pour faire face aux limitations du schéma sécurisé proposé, nous prévoyons d'élargir le champ de notre investigation à divers environnements réseaux, notamment aux réseaux M2M à haute mobilité, y compris les réseaux véhiculaires autonomes, et aux systèmes embarqués hétérogènes dans les réseaux de zone de dispositifs M2M.

La tendance des communications M2M futures est à la coexistence de différents réseaux hétérogènes 5G et 6G , qui utilisent une variété de technologies de transmission impliquant des appareils dont les capacités de communication, de traitement et de stockage variées. Cela encourage à attribuer des tâches distinctes aux appareils (ou machines) connectés, notamment au niveau routage dans le réseau des dispositifs M2M, pour répondre aux différents challenges cités plus haut, et particulièrement à celui de la sécurité, dans le contexte d'un environnement contraint en ressources, et exposé aux attaques physiques et logiques. L'ordonnancement et la distribution des tâches en fonction de leur importance et de leur volume sur les appareils M2M dans le domaine des dispositifs M2M représente aussi un sujet d'intérêt à investir.

Enfin, les tests et les expérimentations que nous avons conduits par simulation ne sauraient remplacer une analyse de scénarios réalistes, dans un environnement le plus proche possible du réel. Afin de compléter les résultats obtenus, il serait intéressant de déployer un réseau de dispositifs M2M avec une blockchain installée sur des stations de base au niveau edge, en utilisant une plateforme telle que Hyperledger Fabric. Cette approche permettrait d'évaluer les mécanismes proposés dans des scénarios réels, en tenant compte des contraintes spécifiques.

BIBLIOGRAPHIE

[1] ETSI TS 102 689 V2.1.1 (2013-07)., Machine-to-Machine communications (M2M); 'M2M service requirements' https://www.etsi.org/deliver/etsi_ts/102600_102699/102689/02.01.01_60/ts_102689v020101p.pdf

[2] J. Kim, J. Lee, J. Kim, and J. Yun, "M2M service platforms: Survey, issues, and enabling technologies," IEEE Commun. Surveys &Tutorials, vol. 16, no. 1, pp. 61–76, 2014, doi: 10.1109/SURV.2013.100713.00203.

[3] Jorge Granjal, Edmundo Monteiro, and Jorge Sá SilvaS, Security Issues and Approaches on Wireless M2M Systems Khan and A.-S.K. Pathan (Eds.): Wireless Networks and Security, SCT, pp. 133–164. DOI: 10.1007/978-3-642-36169-2_5 © Springer-Verlag Berlin Heidelberg 2013

[4] Gurkan Tuna et al, 'A survey on information security threats and solutions for Machine to Machine (M2M) communications', J. Parallel Distrib. Comput. Elsevier, n°:109 (2017), pp-142–154

[5] Manikandan A.1,*, Gayathri Narayanan1, K. S. Reddy Banu Prakash1, Yugandhar Reddy Investigations in Security Challenges and Solutions for M2M Communications—A Review, International Journal of Electrical and Electronic Engineering & Telecommunications Vol. 13, No. 1, 2024

[6] Minhaj Ahmad Khan , Khaled Salah aIoT security: Review, blockchain solutions, and open challenges, Future Generation Computer Systems, Elsevier, n°: 82 (2018), pp-395–411

[7] X. Yun et al., 'An Overview of Blockchain Security Analysis', CCIS 970, pp. 55–72, 2019

[8] Zhang, Y.; Luo, Y.; Chen, X.; Tong, F.; Xu, Y.; Tao, J.; Cheng, G. A Lightweight Authentication Scheme Based on Consortium Blockchain for Cross-Domain IoT. Secur. Commun. Netw. 2022, 2022, 1–15, 10.1155/2022/9686049.

[9] Manzoor, A.; Liyanage, M.; Braeke, A.; Kanhere, S.S.; Ylianttila, M. Blockchain based Proxy Re-Encryption Scheme for Secure IoT Data Sharing. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea, 14–17 , May 2019.

[10] AVISPA. SPAN, the Security Protocol Animator for AVISPA, September 2019. http://www.avispa-project.org/ (accessed on 15 July 2024)

[11] https://www.techtarget.com/iotagenda/definition/machine-to-machine-M2M

[12] K. Stouffer, J. Falco, K. Kent, "Guide to Supervisory Control and data Acquisition (SCADA) and Industrial Control Systems Security," National Institute of Standards and Technology, Tech. Rep, Sept. 2006

[13] Global ICT Standardization Forum for India (GISFI), https://www.gisfi.org/wg_documents/GISFI_IoT_2010097.pdf (accessed on 15 July 2024)

[14] Resul Das, Gurkan Tuna, 'Machine-to-Machine Communications for Smart Homes', International Journal of Computer Networks and Applications (IJCNA) Volume 2, Issue 4, July – August (2015)

[15] Chen, Kwang-Cheng & Lien, Shao-Yu. Machine-to-machine communications: Technologies and challenges. Ad Hoc Networks. 18. 3–23. (2014). http://dx.doi.org/10.1016/j.adhoc.2013.03.007

[16] Chen, M.; Wan, J.; Li, F. Machine-to-machine communications: Architectures, standards, and applications. KSII Trans. Internet Info. Syst. 2012, 6, 480–497.

[17] Yan Zhang, Rong Yu, Shengli Xie, Wenqing Yao, Yang Xiao, M. Guizani, Home M2M networks: Architectures, standards, and QoS improvement, IEEE Commun.Mag. 49 (4) (2011) 44–52.

[18] https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[19] ETSI, Machine-to-machine communications (M2M); definitions (ETSI TR 102 725 V1.1.1, June 2013). Available online: http://www.etsi.org/deliver/etsi_tr/102700_102799/102725/01.01.01_60/tr_102725v010101p.pdf

[20] P. K. Verma, R. Verma, A. Prakash, A. Agrawal, K. Naik, R. Tripathi, M. Alsabaan, T. Khalifa, T. Abdelkader, and A. Abogharaf, "Machine-tomachine (m2m) communications: A survey," Journal of Network and Computer Applications, vol. 66, issue C, pp. 83-105, March 2016.

[21] N. Pandey, M2M communication concept, White Paper, 2016 https://www.researchgate.net/publication/291337307_M2M_communication_concept

[22] A. Elmangoush, A. Al-Hezmi, and T. Magedanz, “The development of M2M standards for ubiquitous sensing service layer,” in 2014 IEEE Globecom Workshops (GC Wkshps), Dec. 2014, pp. 624–629. doi: 10.1109/GLOCOMW.2014.7063502.
[23] M. Ptiček, V. Čačković, M. Pavelić, M. Kušek, and G. Ježić, “Architecture and functionality in M2M standards,” 2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2015 - Proc., no. May, pp. 413–418, 2015, doi: 10.1109/MIPRO.2015.7160306.
[24] Singh, Garima & Shrimankar, Deepti. (2018). Dynamic Group Based Efficient Access Authentication and Key Agreement Protocol for MTC in LTE-A Networks. Wireless Personal Communications. 101. 10.1007/s11277-018-5719-0.
[25] Z. Shelby, K. Hartke, C. Bormann, B. Frank, “Constrained Application Protocol (CoAP),” draft-ietf-core-coap-11; July 2012.
[26] https://www.onem2m.org/harmonization-m2m; OneM2MPartners, oneM2M-TR-0008-Security-V1.0.0 , 2014-April-10.
[27] IEEE 802.16’s Machine-to-Machine (M2M) Task Group, http://www.wirelessman.org/m2m/index.html
[28] WiMAX Forum, www.wimaxforum.org/
[29] IEEE 802.15 WPAN Task Group 4 (TG4): http://www.ieee802.org/15/pub/TG4.html/
[30] ZigBee Alliance: http://www.zigbee.org
[31] M. Gabriel, K. Nandakishore, H. Jonathan, and C. David. “Transmission of IPv6 Packets over IEEE 802.15.4 Networks.” IETF, RFC 4944, September 2007.
[32] ZigBee Alliance, http://www.zigbee.org (accessed on 15 July 2024)
[33] OneM2M, “OneM2M Technical Specification TS-0002-Requirements-V1_0_1,” vol. 1. pp. 1–18, 2014. Disponible sur: https://www.onem2m.org/technical/published-specifications/release-1
[34] Amar Deol, Ken Figueredo, Shi-Wan Lin, Brett Murphy, Dale Seed, Jason Yin, Advancing the Industrial Internet of Things, Editors Shi-Wan Lin and Ken Figueredo, An Industrial Internet Consortium and oneM2M™ Joint Whitepaper. 2019-12-12, oneM2M: Standards for M2M and the IoT: http://onem2m.org/
[35] Open Mobile Alliance OMA-TS-LightweightM2M-V1_0_2-20180209-A, Lightweight Machine to Machine Technical Specification, Approved Version 1.0.2 – 09 Feb 2018
[36] 3GPP TS 22.368 version 13.1.0 Release 13 1 ETSI TS 122 368 V13.1.0 (2016-03). ‘Service Requirements for Machine-Type Communications (MTC)’
[37] F. Ghavimi and H. H. Chen, "M2M Communications in 3GPP LTE/LTE-A Networks: Architectures, Service Requirements, Challenges, and Applications," IEEE Communications Surveys Tutorials, 2014.
[38] Dmitry Namiot, Manfred Sneps-Sneppe, ‘On M2M Software Platforms’, International Journal of Open Information Technologies ISSN: 2307-8162 vol. 2, no. 8, 2014
[39] Global M2M Platform Market By Type (GSM, GPRS), By Application (Transport, Energy), By Geographic Scope And Forecast, Report ID : 507201, February 2023, Disponible online sur: https://www.verifiedmarketreports.com/download-sample/?rid=507201
[40] https://www.ptc.com/en/products/thingworx (accessed on 15 July 2024)
[41] https://m2msupport.net/m2msupport//att-m2m-modules-certification-data-plans-service-platform/ (accessed on 15 July 2024)
[42] http://www.etherios.com/ (accessed on 15 July 2024)
[43] https://ec.eurotech.com/ (accessed on 15 July 2024)
[44] https://sensorcloud.com/ (accessed on 15 July 2024)
[45] https://www.digimarc.com/?utm_source=evrythng (accessed on 15 July 2024)
[46] https://www.arduino.cc/ (accessed on 15 July 2024)
[47] P. Masek, J. Hosek, and M. Dubrava, “Influence of M2M communication on LTE networks,” in the 10th International IEEE Conference, Zvule, Czech Republic, August 2014.
[48] D. Kovac, P. Masek, and J. Hosek, “Simulation-Based Study on Capacity Performance of 4G Mobile Network for M2M Services,” in the International Conference "Technical Universities: Integration with European and World Systems of Education", Izhevsk, Russia, April 2014.

[49] Imani Amirhossein & Keshavarz-Haddad, Alireza & Eslami, Mohsen & Haghighat, Javad. Security Challenges and Attacks in M2M Communications. 9th International Symposium on Telecommunications (IST'2018), 264-269. (2018).DOI:10.1109/ISTEL.2018.866104

[50] Y.-C. Chang, "Study of Overload Control Problem for Intelligent LTE M2M Communication System," Advances in Smart Systems Research, vol. 3, no. 3, pp. 44–48, 2013.

[51] C. Yu, K. Doppler, C. Ribeiro, and O. Tirkkonen, "Resource sharing optimization for device-to-device communication underlaying cellular networks," IEEE Trans. Wireless Commun., vol.10, Aug. 2011, pp.2752-2763.

[52] M. Zulhasnine, C. Huang, and A. Srinivasan, "Efficient resource allocation for device-to-device communication underlaying LTE network," IEEE 6th Int. Conf. on Wireless and Mobile Computing, Networking and Communications, 11-13 Oct. 2010, pp.368-375.

[53] S. Zrncic, I. Bojic, D. Katusic, P. Skocir, M. Kusek and G. Jezic, "Quality-of-Service in Machine-to-Machine service provisioning process," 2013 21st International Conference on Software, Telecommunications and Computer Networks - (SoftCOM 2013), Split, Croatia, 2013, pp. 1-5, doi: 10.1109/SoftCOM.2013.6671899. DOI:10.1109/SoftCOM.2013.6671899

[54] Asma Elmangousha*, Andreea Ancuta Coricib , Ronald Steinkeb , Marius Coricib , Thomas Magedanza,, 'A Framework for Handling Heterogeneous M2M Traffic', The 6th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2015), Procedia Computer Science 63 (2015) Elsevier B.V. 112 – 119 1877-0509 © 2015

[55] Alam, Mohammed Jaber & Hossain, Rahat & Azad, Salahuddin & Chugh, Ritesh. (2023). An overview of LTE/LTE‐A heterogeneous networks for 5G and beyond. Transactions on Emerging Telecommunications Technologies. 34(1)- June 2023 DOI:10.1002/ett.4806,

[56] Li Liao, Chengjun Ji, "Wireless Resource Management and Resilience Optimization of the M2M-Oriented Mobile Communication System", Journal of Sensors, vol. 2021, Article ID 9596606, 11 pages, 2021. https://doi.org/10.1155/2021/9596606

[57] Pradhan, Devasis & Tun, Hla. 'Security Challenges: M2M Communication in IoT. ' Journal of Electrical Engineering and Automation 4(3):187-199 October 2022. DOI:10.36548/jeea.2022.3.006

[58] H. Verma, N. Chauhan, and L. K. Awasthi, "A comprehensive review of 'internet of healthcare things': Networking aspects, technologies, services, applications, challenges, and security concerns," Comput. Sci. Rev., vol. 50, 2023. doi: 10.1016/j.cosrev.2023.10059

[59] M. M. Mogadem, Y. Li, and D. L. Meheretie, "A survey on internet of energy security: related fields, challenges, threats and emerging technologies," Cluster Comput., vol. 25, no. 4, 2022. doi: 10.1007/s10586-021-03423-z

[60] U. Singh, A. Dua, N. Kumar et al., "Scalable priority-based resource allocation scheme for M2M communication in LTE/LTE-A network," Computers and Electrical Engineering, vol. 103, 2022. doi: 10.1016/j.compeleceng.2022.108321

[61] G. Agosta, A. Barenghi and G. Pelosi, "Securing software cryptographic primitives for embedded systems against side channel attacks", International Carnahan Conference on Security Technology (ICCST), Rome,pp. 1-6, 2014.

[62] Méndez Real M, Salvador R. Physical Side-Channel Attacks on Embedded Neural Networks: A Survey. Applied Sciences. 2021; 11(15):6790. https://doi.org/10.3390/app11156790

[63] Y. K. Saheed, A. I. Abiodun, S. Misra et al., "A machine learning-based intrusion detection for detecting internet of things network attacks," Alexandria Engineering Journal, vol. 61, no. 12, 2022. doi: 10.1016/j.aej.2022.02.063

[64] R. K. Shrivastava, S. P. Singh, M. K.l Hasan et al., "Securing internet of things devices against code tampering attacks using Return Oriented Programming," Comput. Commun., vol. 193, 2022. doi: 10.1016/j.comcom.2022.06.033

[65] M. Jalalitabar, M. Valero and A. G. Bourgeois, "Demonstrating the Threat of Hardware Trojans in Wireless Sensor Networks", 24th International Conference on Computer Communication and Networks (ICCCN), Las Vegas, NV, pp. 1-8 , 2015.

[66] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, “Denial of service attacks in wireless networks: The case of jammers,” IEEE Communications Surveys and Tutorials, vol. 13, no. 2. 2011. doi: 10.1109/SURV.2011.041110.00022

[67] W. Dong and X. Liu, “Robust and Secure Time-Synchronization Against Sybil Attacks for Sensor Networks,”, IEEE Transactions on Industrial Informatics, vol. 11, no. 6, pp. 1482-1491, Dec. 2015.

[68] T L Kao1, H C Wang, J E Li, ‘Safe MQTT-SN: a lightweight secure encrypted communication in IoT Journal of Physics: Conference Series 2020 (2021) 012044 doi:10.1088/1742-6596/2020/1/012044

[69] Muhammad Imran Malik, , Ian Noel, , Peter Hannay, Syed Naeem Firdous,, Zubair Baig McAteer XMPP architecture and security challenges in an IoT ecosystem, Proceedings of the 16th Australian Information Security Management Conference (2018)

[70] M. Singh, M. A. Rajan, V. L. Shivraj and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 2015, pp. 746-751, doi: 10.1109/CSNT.2015.16.

[71] N. Naik, “Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP,” 2017. doi: 10.1109/SysEng.2017. 8088251.

[72] Victor Seoane, Carlos Garcia-Rubio, Florina Almenares, Celeste Campo,’ ‘Performance evaluation of CoAP and MQTT with security support for IoT environments‘ Computer Networks, Volume 197, 2021, 108338, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2021.108338

[73] Nastase, L. (2017), ‘ Security in the Internet of Things: A survey on application layer protocols’, Control Systems and Computer Science (CSCS), 2017 21st International Conference on, Bucharest, Romania. http://ieeexplore.ieee.org/document/7968629/

[74] Coruh, U.; Bayat, O. Hybrid Secure Authentication and Key Exchange Scheme for M2M Home Networks. Secur. Commun. Netw. 2018, 2018, 6563089. [Google Scholar] [CrossRef]

[75] Lara, E.; Aguilar, L.; Sanchez, M.A.; García, J.A. García Lightweight Authentication Protocol for M2M Communications of Resource-Constrained Devices in Industrial Internet of Things. Sensors 2020, 20, 501. [Google Scholar] [CrossRef] [PubMed] [Green Version]

[76] Bogdanov, A.; Knudsen, L.R.; Leander, G.; Paar, C.; Poschmann, A.; Robshaw, M.J.B.; Seurin, Y.; Vikkelsoe, C. Present: An ultra-lightweight block cipher. In International Workshop on Cryptographic Hardware and Embedded Systems; Springer: Berlin/Heidelberg, Germany, 2007; pp. 450–466. [Google Scholar]

[77] Hong, D.; Lee, J.-K.; Kim, D.-C.; Kwon, D.; Ryu, K.H.; Lee, D.-G. LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors. In Information Security Applications; Springer: Berlin/Heidelberg, Germany, 2014; pp. 3–27. [Google Scholar]

[78] Bae, Gi-chur & Shin, Kyung-wook. (2016). An Efficient Hardware Implementation of Lightweight Block Cipher Algorithm CLEFIA for IoT Security Applications. Journal of the Korea Institute of Information and Communication Engineering. 20. 351-358. 10.6109/jkiice.2016.20.2.351.

[79] Seok, Byoungjin & Park, Jinseong & Park, Jong. (2019). A Lightweight Hash-Base Blockchain Architecture for Industrial IoT. Applied Sciences. 9. 3740. 10.3390/app9183740.

[80] Schukat, Michael & Cortijo, Pablo. (2015). Public key infrastructures and digital certificates for the Internet of things. 10.1109/ISSC.2015.7163785.

[81] Fakroon, M.; Alshahrani, M.; Gebali, F.; Traore, I. Secure remote anonymous user authentication scheme for smart home environment. Internet Things 2020, 9, 100158. .

[82] Hsin Chung Chen, Mohammad Abdullah Al Faruque, Pai H. Chou,’ Security and Privacy Challenges in IoT-Based Machine-to-Machine Collaborative Scenarios’, 2016, CODES/ISSS ’16, October 01-07, 2016, Pittsburgh, PA, USA ACM. ISBN 978-1-4503-4483-8/16/10 DOI: http://dx.doi.org/10.1145/2968456.2974008

[83] Perrig, A.; Szewczyk, R.; Tygar, J.D.; Culler, D.E. SPINS: Security Protocols for Sensor Networks. Wirel. Netw. 2002, 8, 521–534.

[84] Karlof, C.; Sastry, N.;Wagner, D. TinySec: A link layer security architecture for wireless sensor networks. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys ‘04, Baltimore, MD, USA, 3–5 November 2004; Association for Computing Machinery: New York, NY, USA, 2004; pp. 162–175.

[85] Luk, M.; Mezzour, G.; Perrig, A.; Gligor, V. MiniSec: A Secure Sensor Network Communication Architecture. In Proceedings of the 2007 6th International Symposium on Information Processing in Sensor Networks, Cambridge, MA, USA, 25–27 April 2007.

[86] Chen, Yu-Wen & Wang, Jui-Tang & Chi, Kuang-Hui & Tseng, Chien-Chao. (2012). Group-Based Authentication and Key Agreement. Wireless Personal Communications. 62. 965-979. 10.1007/s11277-010-0104-7.

[87] Chengzhe Lai a,b , Hui Li a , Rongxing Lu c , Xuemin (Sherman) Shen b, SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks / Computer Networks 57 (2013) 3492–3510.

[88] Giustolisi, Rosario & Gehrmann, Christian & Ahlström, Markus & Holmberg, Simon. (2017). A Secure Group-Based AKA Protocol for Machine-Type Communications. Lecture Notes in Computer Science. 10157. 3-27. 10.1007/978-3-319-53177-9_1.

[89] Lai, Chengzhe & Lu, Rongxing & Zheng, Dong & Li, Hui & Shen, Xuemin. (2016). GLARM: Group-based Lightweight Authentication Scheme for Resource-constrained Machine to Machine Communications. Computer Networks. 99. 10.1016/j.comnet.2016.02.007.

[90] Evangelina Lara, Leocundo Aguilar, Mauricio A. Sanchez and Jesús A. García, Lightweight Authentication Protocol for M2M, Communications of Resource-Constrained Devices in Industrial Internet of Things, Sensors 2020, 20, 501; doi:10.3390/s20020501

[91] Alireza Esfahani et al. A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment. IEEE Internet of Things n°: 6,1-2019,Page(s):288 – 296

[92] Qiu, Y.; Ma, M. A Mutual Authentication and Key Establishment Scheme for M2M Communication in 6LoWPAN Networks. IEEE Trans. Ind. Inform. 2016, 12, 2074–2085. DOI:10.1109/JIOT.2017.2737630

[93] Gao, L.; Zhang, L.; Feng, L.; Ma, M. An Efficient Secure Authentication and Key establishment Scheme for M2M Communication. in 6LoWPAN in Unattended Scenarios. Wirel. Pers. Commun. 2020, 115, 1603–1621.

[94] Hussen, H.R.; Tizazu, G.A.; Ting, M.; Lee, T.; Choi, Y.; Kim, K.-H. Sakes: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6LoWPAN). In Proceedings of the 2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN), Da Nang, Vietnam, 2–5 July 2013; pp. 246–251.

[95] Xiao L,Wan X, Lu X, Zhang Y,Wu D. IoT security techniques based on machine learning: how do IoT devices use AI to enhance security? IEEE Signal Process Mag. 2018;35(5):41-49.

[96] Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P. Network intrusion detection for IoT security based on learning techniques. IEEE Commun Surv Tutor. 2019;21(3):2671-2701.

[97] Anthi E,Williams L, Słowinska M, Theodorakopoulos G, Burnap P. A supervised intrusion detection system for smart home IoT devices. IEEE Internet Things J. 2019;6(5):9042-9053.

[98] Bhabendu Kumar Mohanta, Debasish Jena, Utkalika Satapathy, Srikanta Patnaik, Survey on IoT Security:Challenges and Solution using Machine Learning, Artificial Intelligence and Blockchain Technology, Internet of Things (2020), doi: https://doi.org/10.1016/j.iot.2020.100227

[99] Sagu A, Gill NS. Machine learning techniques for securing IoT environment. Int J Innov Technol Explor Eng. 2020;9(4):978-982.

[100] Haber, S.; Stornetta, W.S. How to time-stamp a digital document. In Advances in Cryptology-CRYPTO' 90, Proceedings of the Conference on the Theory and Application of Cryptography, Santa Barbara, CA, USA, 11–15 August 1990; Springer: Berlin/Heidelberg, Germany, 1990

[101] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. Decentralized Bus. Rev. 2008, 21370. Available online: https://www.ussc.gov/sites/default/files/pdf/training/annual-national-training-seminar/2018/Emerging_Tech_Bitcoin_Crypto.pdf (accessed on 15 July 2024).

[102] Cheng, S., Zeng, B. and Huang, Y.Z. (2017), "Research on application model of blockchain technology in distributed electricity market"

[103] S. Seibold and G. Samman, “Consensus: Immutable agreement for the Internet of value,” KPMG¡ https://assets.kpmg.com/content/dam/kpmg/pdf/2016/06/kpmg-blockchain-consensus-mechanism.pdf?aid=fndrabg_p?aid=fndrabg_p, 2016
[104] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, “A survey of distributed consensus protocols for blockchain networks,” IEEE Commun Surv Tutor., vol. 22, no. 2, pp. 1322–1465, 2020, doi: 10.1109/COMST.2020.2969706
[105] Architecture for Blockchain Applications 2019 - Xiwei Xu, Ingo Weber, Mark Staples
[106] Ethereum : WOOD, Gavin, et al. Ethereum: A secure decentralized generalised transaction ledger. Ethereum project yellow paper, 2014, vol. 151, no 2014, p. 1-32.
[107] Sunny King: Primecoin: Cryptocurrency with Prime Number Proof-of-Work (2013) https://primecoin.io/bin/primecoin-paper.pdf
[108] An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends - 2017
[109] Proof of stake instead of proof of work (2011) https://bitcointalk.org/index.php?topic=27787.0
[110] Network, Fast, Cheap, Scalable Token Transfers for Ethereum. Available online: https://github.com/raiden-network/raiden (accessed on 19 July 2018).
[111] Hu, Junjie & Liu, Ke. (2020). Raft consensus mechanism and the applications. Journal of Physics: Conference Series. 1544. 012079. 10.1088/1742-6596/1544/1/012079.
[112] An, Min & Fan, Qiyuan & Yu, Hao & An, Bo & Wu, Nannan & Zhao, Haiyang & Wan, Xinhao & Li, Jiaxuan & Wang, Rui & Zhen, Jingyu & Zou, Qinyan & Zhao, Bin. (2023). Blockchain Technology Research and Application: A Literature Review and Future Trends. Journal of Data Science and Intelligent Systems. 10.47852/bonviewJDSIS32021403.
[113] M. Niranjanamurthy, B. N. Nithya, and S. Jagannatha, “Analysis of Blockchain technology: pros, cons and SWOT,” Clust. Comput., vol. 22, no. 6, pp. 14743–14757, 2019, doi: 10.1007/s10586-018-2387-5.
[114] Blockchain Blueprint for a new economy (2015) - Melanie Swan
[115] https://www.hyperledger.org/ (accessed on 15 July 2024)
[116] https://www.corda.net/. (accessed on 15 July 2024)
[117] https://ripple.com/. (accessed on 15 July 2024)
[118] https://z.cash/. (accessed on 15 July 2024)
[119] Divya M., Nagaveni B. Biradar (2018): OTA-Next Generation Blockchain
[120] https://stellar.org/ (accessed on 15 July 2024)
[121] https://tezos.com/ (accessed on 15 July 2024)
[122] https://cardano.org/ (accessed on 15 July 2024)
[123] Srivastava A., Bhattacharya P., Singh A., Mathur A., Prakash O., Pradhan R. 2018: A Distributed Credit Transfer Educational Framework based on Blockchain
[124] Foroglou, G., Tsilidou, A.L. (2015): Further applications of the blockchain
[125] Springer Nature Switzerland AG 2018 K. Yoshida and M. Lee (Eds.): PKAW 2018, LNAI 11016, pp. 201–210, 2018. https://doi.org/10.1007/978-3-319-97289-3_15
[126] https://fr.statista.com/themes/9325/les-cryptomonaies/ (accessed on 15 July 2024)
[127] N. Szabo, “Smart contracts: building blocks for digital markets,” EXTROPY J. Transhumanist Thought16, vol. 18, no. 2, p. 28, 1996.
[128] S. N. Khan, F. Loukil, C. G. Guegan, E. Benkhelifa, and A. BaniHani, “Blockchain smart contracts: Applications, challenges, and future trends,” Peer–Peer Netw Appl, vol. 14, no. 5, pp. 2901–2925, 2021, doi: 10.1007/s12083-021-01138-0.
[129] H. Huang, K.-C. Li, and X. Chen, “Blockchain-based fair three-party contract signing protocol for fog computing,” Concurr. Comput. Pract. Exp., vol. 31, no. 22, 2019, doi: 10.1002/cpe.4469
[130] Khan, Dodo, Low Tang Jung, and Manzoor Ahmed Hashmani. 2021. "Systematic Literature Review of Challenges in Blockchain Scalability" Applied Sciences 11, no. 20: 9372. https://doi.org/10.3390/app11209372
[131] https://bytesoft.vn/en/what-is-blockchain-4-0 (accessed on 15 July 2024)
[132] Zhou, Q.; Huang, H.; Zheng, Z.; Bian, J. Solutions to Scalability of Blockchain: A Survey. IEEE Access 2020, 8, 16440–16455.

[133] Kim, S.; Kwon, Y.; Cho, S. A Survey of Scalability Solutions on Blockchain. In Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 17–19 October 2018; pp. 1204–1207.
[134] Hafid, A.; Hafid, A.S.; Samih, M. Scaling Blockchains: A Comprehensive Survey. IEEE Access 2020, 8, 136244–136262.
[135] Del Monte, G.; Pennino, D.; Pizzonia, M. Scaling Blockchains without Giving Up Decentralization and Security. arXiv 2020, arXiv:2005.06665. Available online: https://arxiv.org/abs/2005.06665 (accessed on 17 June 2021).
[136] C. Ye, G. Li, H. Cai, Y. Gu, and A. Fukuda, "Analysis of security in blockchain: Case study in 51%-attack detecting," 2018 5th International conference on dependable systems and their applications (DSA), IEEE, pp. 15–24, 2018.
[137] R. Gupta, A. Kumari, and S. Tanwar, "A taxonomy of blockchain envisioned edge-as-a-connected autonomous vehicles," Transactions on Emerging Telecommunications Technologies, vol. 32, no. 6, p. e4009, 2021.
[138] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," 2019 International Conference on Computing, Networking and Communications (ICNC), pp. 360–364, 2019.
[139] Hsiao, H.-I.; Huang, K.-L. Time-temperature transparency in the cold chain. Food Control 2016, 64, 181–188.
[140] Marc Pilkington (2016): Blockchain Technology: Principles and Applications
[141] DHL Trend Research (2018): BLOCKCHAIN IN LOGISTICS: Perspectives on the upcoming impact of blockchain technology and use cases for the logistics industry
[142] Republic of Estonia E-Residency, "The new digital nation" [online]. Available from: https://e-resident.gov.ee/ (accessed on 15 July 2024)
[143] Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," IEEE Internet Things J., vol. 5, no. 2, pp. 1184–1195, 2018.
[144] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in 2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops), IEEE, 2017, pp. 618–623.
[145] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," IEEE Trans. Ind. Inform., vol. 14, no. 8, pp. 3690–3700, 2018, doi: 10.1109/TII.2017.2786307.
[146] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," IEEE Commun. Mag., vol. 56, no. 8, pp. 33–39, 2018.
[147] Lazrag, H.; Chehri, A.; Saadane, R.; Rahmani, M.D. A Blockchain-Based Approach for Optimal and Secure Routing in Wireless Sensor Networks and IoT. In Proceedings of the 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Sorrento, Italy, 26–29 November 2019; pp. 411–415. https://doi.org/10.1109/SITIS.2019.00072.
[148] Islam, A.; Shin, S.Y. BUAV: A Blockchain Based Secure UAV-Assisted Data Acquisition Scheme in Internet of Things. J. Commun. Netw. 2019, 21, 491–502.
[149] Manzoor, A.; Liyanage, M.; Braeke, A.; Kanhere, S.S.; Ylianttila, M. Blockchain based Proxy Re-Encryption Scheme for Secure IoT Data Sharing. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea, 14–17 May 2019.
[150] Sikeridis, D.; Bidram, A.; Devetsikiotis, M.; Reno, M.J. Blockchain-Based Mechanism for Secure Data Exchange in Smart Grid Protection Systems. In Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 10–13 January 2020.
[151] Bouras, M.A.; Lu, Q.; Dhelim, S.; Ning, H. A Lightweight Blockchain-Based IoT Identity Management Approach. Future Internet 2021, 13, 24. https://doi.org/10.3390/fi13020024.
[152] Gangwani, P.; Perez-Pons, A.; Bhardwaj, T.; Upadhyay, H.; Joshi, S.; Lagos, L. Securing Environmental IoT Data Using Masked Authentication Messaging Protocol in a DAG-Based Blockchain: IOTA Tangle. Future Internet 2021, 13, 312. https://doi.org/10.3390/fi13120312.

[153] Saqib, M.; Jasra, B.; Moon, A.H. A lightweight three factor authentication framework for IoT based critical applications. J. King Saud Univ. Comput. Inf. Sci. 2021. ISSN 1319-1578.
[154] Challa, S.; Wazid, M.; Das, A.K.; Kumar, N.; Reddy, A.G.; Yoon, E.-J.; Yoo, K.-Y. Secure signature-based authenticated key establishment scheme for future IoT applications. IEEE Access 2017, 5, 3028–3043.
[155] Li, X.; Peng, J.; Niu, J.; Wu, F.; Liao, J.; Choo, K.-K.R. A robust and energy efficient authentication protocol for industrial internet of things. IEEE Internet Things J. 2017, 5, 1606–1615.
[156] Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 7 January 2000.
[157] Bilami, Karam & LORENZ, Pascal. (2022). Lightweight Blockchain-Based Scheme to Secure Wireless M2M Area Networks. Future Internet. 14. 158. 10.3390/fi14050158.
[158] Thomas Hardjono, Ned Smith, 'Cloud-Based Commissioning of Constrained Devices using Permissioned Blockchains Proceedings of ACM IoT Privacy, Trust & Security - IoTPTS 2016, Xi'an, China, May 2016.
[159] Meryem Ammi, Shatha Alarabi, and Elhadj Benkhelifa. Customized blockchain-based architecture for secure smart home for lightweight iot. Information Processing & Management, 58(3):102482, 2021.
[160] Karam Eddine Bilami , Jaafar Gaber, Pascal Lorenz "Blockchain-based Authentication Protocol for Wireless M2M area Networks with Sidechain Integration", 8th IEEE Symposium on Wireless Technology & Applications (ISWTA 2024), 20-21 July 2024, Kuala Lumpur, Malaysia 2024.
[161] Low Energy Adaptive Clustering Hierarchy protocol (LEACH) (https://www.mathworks.com/matlabcentral/fileexchange/44073-low-energy-adaptive-clustering-hierarchy-protocol-leach).
[162] Shibasaki, Y.; Iwamura, K.; Sato, K. A Communication-Efficient Secure Routing Protocol for IoT Networks. Sensors 2022, 22, 7503. https://doi.org/10.3390/s22197503
[163] G. Zapata Secure 'Ad hoc On-Demand Distance Vector Routing', Mobile Computing and Communications Review, Volume 6, Number 3.
[164] Mansour, I.; Rusinek, D.; Chalhoub, G.; Lafourcade, P.; Ksiezopolski, B. Multihop Node Authentication Mechanisms for wireless Sensor Networks. In Proceedings of the 13th International Conference (ADHOC-NOW 2014), Benidorm, Spain, 22–27 June 2014
[165] S. Othmen, F. Zarai, A. Belghith and L. Kamoun, "Anonymous and secure on-demand routing protocol for multi-hop cellular networks," 2016 International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, Tunisia, 2016,
[166] Lutful Karim, Alagan Anpalagan, Nidal Nasser, Jalal Almhana and Isaac Woungang. 'Fault tolerant energy efficient and secure clustering scheme for mobile machine-to-machine communications', Transactions on Emerging Telecommunications Technologies. 2014; 25:pp. 1028–1044 Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/ett.2801
[167] M. Schmittner, A. Asadi and M. Hollick, "SEMUD: Secure multi-hop device-to-device communication for 5G public safety networks," 2017 IFIP Networking Conference (IFIP Networking) and Workshops, Stockholm, Sweden, 2017, pp. 1-9, doi: 10.23919/IFIPNetworking.2017.8264846.
[168] Salwa Othmen, Wahida Mansouri, Somia Asklany and Wided Ben Daoud, "Optimize and Secure Routing Protocol for Multi-hop Wireless Network" International Journal of Advanced Computer Science and Applications(IJACSA), 13(1), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130130

Liste des publications

Bilami, Karam & LORENZ, Pascal. (2022). Lightweight Blockchain-Based Scheme to Secure Wireless M2M Area Networks. Future Internet. 14. 158. 10.3390/fi14050158.

Karam Eddine Bilami , Jaafar Gaber, Pascal Lorenz “Blockchain-based Authentication Protocol for Wireless M2M area Networks with Sidechain Integration”, 8th IEEE Symposium on Wireless Technology & Applications (ISWTA 2024), 20-21 July 2024, Kuala Lumpur, Malaysia 2024.